

AI CONFINI DELLO SPECTRUM

APPLICAZIONI AVANZATE

P. Williams



EDIZIONE ITALIANA



GRUPPO
EDITORIALE
JACKSON

AI CONFINI DELLO SPECTRUM

APPLICAZIONI AVANZATE

P. Williams



GRUPPO
EDITORIALE
JACKSON
Via Rosellini, 12
20124 Milano

Publicato negli U.S.A. da:

Melbourne House Software Inc. - 233 South Beverly Drive, Beverly Hills, California 90212

Publicato nel Regno Unito da: Melbourne House (Editori) Ltd., Glebe Cottage, Glebe House, Station Road, Cheddington, Leighton Buzzard, Bedfordshire. W7,7 NA. ISBN 0 86161 109 8

Publicato in Australia da: Melbourne House (Australia) Pty. Ltd., Suite 4, 75 Palmerston Crescent, South Melbourne, Victoria, 3205. National Library of Australia Card Number e ISBN 0 86759 112 9.

Tutti i programmi in questo libro sono Copyright (c) 1982. I programmi, quando non altrimenti indicato, sono Copyright (c) 1982 by Beam Software.

Questa edizione è pubblicata per accordi con la Melbourne House (Publisher) Ltd London.

© Copyright per l'edizione italiana: Gruppo Editoriale Jackson - Febbraio 1985

SUPERVISIONE TECNICA: Daria Gianni

GRAFICA E IMPAGINAZIONE: Francesca Di Fiore

TRADUTTORE: Luca Brigatti

COPERTINA: Silvana Corbelli

FOTOCOMPOSIZIONE: CorpoNove s.n.c. - Bergamo

STAMPA: Grafica '78 - Pioltello

Tutti i diritti sono riservati. Stampato in Italia. Nessuna parte di questo libro può essere riprodotta, memorizzata in sistemi di archivio, o trasmessa in qualsiasi forma o mezzo, elettronico, meccanico, fotocopia, registrazione o altri senza la preventiva autorizzazione scritta dell'editore.

SOMMARIO

PROGRAMMI GENERALI

| | | |
|------------------------------|------|----|
| Saltarana | Pag. | 1 |
| Ordina i numeri | » | 3 |
| Asteroidi nello spazio | » | 5 |
| Orologio Spectrum | » | 7 |
| 3-D Mazeman | » | 13 |

DIDATTICI

| | | |
|-------------------------|---|----|
| Test di geometria | » | 19 |
| Re e Regine | » | 21 |

GIOCHI D'AZZARDO

| | | |
|---------------------|---|----|
| Blackjack | » | 27 |
| Fruit Machine | » | 33 |

MATEMATICI

| | | |
|----------------------------|---|----|
| Bubblesort | » | 45 |
| Equazioni simultanee | » | 51 |

ARCADE GAMES

| | | |
|-------------------------|---|----|
| Fuga spaziale | » | 55 |
| Allunaggio | » | 59 |
| Attacco spaziale | » | 63 |
| Spectrum invaders | » | 67 |
| Meteor storm | » | 73 |
| Eliminator | » | 81 |
| Freeway frog | » | 87 |

UTILITIES

| | | |
|-----------------------------------|---|-----|
| Grafici ad alta risoluzione | » | 99 |
| Line renumbering | » | 107 |
| Block line delete | » | 115 |
| Monitor codice macchina | » | 119 |

PROGRAMMI FINANZIARI

| | | |
|-----------------------------|---|-----|
| Libro paga | » | 127 |
| Analisi delle vendite | » | 135 |
| Valutazione dei beni | » | 141 |

PROGRAMMI DI STRATEGIA

| | | |
|-------------------|---|-----|
| Scacchi | » | 147 |
| Dama | » | 153 |
| L'avventura | » | 165 |

NOTE SUI LISTATI DEI PROGRAMMI

Come sapete, ci sono diversi modi di usare la tastiera dello Spectrum, che dipendono dai simboli, grafici e no, disponibili sullo Spectrum.

Abbiamo listato i programmi di questo libro in un modo particolare per renderli più leggibili; occorre però qualche spiegazione.

▲ Indica uno spazio bianco e lo troverete sia in stringhe di caratteri grafici, sia in normali istruzioni di PRINT.
Ricordiamo che l'esatto uso degli spazi è spesso indispensabile per il buon funzionamento del programma.

Attenzione però! Abbiamo aggiunto questo simbolo solo quando abbiamo ritenuto difficile la lettura dei listati con i normali spazi bianchi. (L'assenza dei triangolini non significa che non ci sono spazi!).

{S} Indica che occorre premere CAPS SHIFT insieme al carattere seguente. Es. {S8}.

{E} Indica che occorre portarsi nel modo esteso per il carattere seguente. Es. {S5}.
Il modo esteso si ottiene premendo insieme CAPS SHIFT e SIMBOL SHIFT.

{G} È usato per indicare che occorre portarsi nel modo grafico. Per esempio {4G8} significa "digitate quattro volte il tasto 8 nel modo grafico" mentre {4SG8} significa "digitate 4 volte il tasto 8 tenendo premuto CAPS SHIFT in modo grafico".

A Le lettere sottolineate indicano l'uso dei caratteri grafici definiti dall'utente. Quando incontrerete queste istruzioni, dovrete semplicemente portarvi nel modo grafico e premere i tasti indicati dalla sottolineatura, ottenendo così dei caratteri speciali.
Ricordatevi poi di tornare nel modo normale!

Nei programmi di questo libro, sono state sfruttate le capacità dello Spectrum di specificare i colori della carta e dell'inchiostro senza ricorrere alle specifiche istruzioni di PAPER e INK.

Si sfrutta questa possibilità disponendosi nel modo esteso e premendo l'appropriato tasto del colore (per la carta) o il tasto CAPS SHIFT e quello del colore (per l'inchiostro).

Per intenderci meglio; {ES3} sceglie il colore magenta per l'inchiostro ed E4 sceglie il colore verde per la carta. Se le due istruzioni sono successive si ottiene la scrittura in magenta su fondo verde. Vi consigliamo di fare qualche prova per abituarvi a questo modo di dare le istruzioni di colore.

In alcuni programmi di questo libro abbiamo scelto di non inserire le istruzioni di colore, ora descritte, per non appesantire eccessivamente i listati.

Potrete comunque vedere i colori scelti per i caratteri direttamente sul video, dai listati dei programmi registrati sulla cassetta.

Se disponete di un TV b/n, non vi sarà difficile trovare da soli i colori più opportuni. Anzi in qualche caso potrete migliorare la visibilità delle scene, che sono state pensate per video a colori.

I programmi in questione sono: Frog caratt, fway frog, eliminator.

Per finire un consiglio importante: prima di caricare un programma di questo libro, spegnete e riaccendete lo Spectrum, oppure date le istruzioni NEW e RANDOMIZE USR 0. Questa operazione però non va fatta fra i programmi "frog caratt" e "f'way frog".

INTRODUZIONE ALL'EDIZIONE ITALIANA

La pubblicazione di *un* libro di programmi sullo Spectrum non è, in sé, un avvenimento editoriale eccezionale.

Per questo che è stato definito il miglior microcomputer del mondo esiste, infatti, una gran letteratura e un libro di programmi non è certo una novità.

Neanche la *quantità* di programmi presentata in questo libro costituisce, invero, alcunché di affatto rimarchevole (questa stessa Casa Editrice ha pubblicato un libro contenente ben 77 programmi per lo Spectrum).

Quello che è notevole, e che fa di questo libro un libro eccezionale, è la *qualità* dei programmi presentati nonché le tecniche di programmazione adottate, un esame attento delle quali permetterà all'utente di migliorare notevolmente la propria *ars* programmatoria iniziandolo ai "segreti" della programmazione strutturata.

La qualità e la varietà del software presentato appaiono evidenti già da un primo esame dell'indice; in questo libro troviamo programmi che vanno dagli arcade più famosi (fra i quali è annoverato il celeberrimo "Spectrum Invaders"!) alle utilities più interessanti (fra le quali un prezioso renumber), dai giochi d'azzardo ai programmi didattici, dai programmi finanziari ai programmi matematici e persino ai programmi di strategia, fra i quali va segnalato un programma per il gioco degli scacchi scritto interamente in BASIC che, pur non essendo il più veloce fra i programmi, illustra efficacemente la potenza del BASIC Spectrum.

Per ogni programma, non ci si è limitati a dare una spiegazione del suo funzionamento, ma è stato dato ampio spazio alla descrizione delle tecniche di programmazione adottate, che saranno utilissime a tutti coloro che non si limitano ad usare programmi già pronti, ma che programmano anche per conto loro.

Dunque, mano allo Spectrum e buon divertimento!

LUCA BRIGATTI

Saltarana

Copyright (c) by Beam Software.

Il gioco "saltarana" è grazioso e semplice.

Si comincia con due opposti gruppi di rane, e ogni rana può solo spostarsi in uno spazio adiacente o saltare un'altra rana.

```
XXXX-0000  
123456789
```

Così, per esempio come prima mossa, la rana in posizione 4 può muovere in 5, o la rana in 6 può muovere in 5, o la rana in 3 può saltare la 4 per atterrare in 5, o la rana in 7 può saltare la 6 e atterrare in 5.

Lo scopo del gioco è cercare di spostare tutte le rane di destra a sinistra e viceversa nel minor numero di mosse. È un gran divertimento!

STRUTTURA DEL PROGRAMMA:

La prima parte di ogni programma serve per inizializzare tutte le variabili che vengono usate. In questo caso, vogliamo definire la posizione iniziale delle rane, e mettere a zero il numero di mosse.

Uno sguardo al programma rivela la seguente struttura:

Print Inizializza le variabili. Stampa la posizione delle rane. Controlla se è finito.
Se sì vai a Fine.

Input Introduci la mossa del giocatore. Controlla se la mossa è consentita.
Se non lo è vai ancora a Input.

Aggiungi 1 al numero di mosse.
Fai la mossa e vai a Print.

Fine Congratulazioni al giocatore. Chiedi al giocatore se vuole giocare ancora.
Se sì allora RUN.

Questo semplice approccio ci fornisce una visione d'insieme del programma e ci consente di capirne la struttura nel caso volessimo successivamente fare dei cambiamenti.

STRUTTURA DELLE VARIABILI:

Per questo programma useremo "variabili stringa" per definire la posizione delle rane.

Una variabile stringa è facile da manipolare in questo contesto e rende molto veloce la stampa.

Definiamo a\$ con la posizione originale delle rane e b\$ con la posizione attuale. Possiamo usare queste stesse variabili per controllare se il programma è terminato (vedi linea 150).

Usiamo le variabili "a" e "da" per rappresentare la posizione di arrivo e di partenza della rana.

Poiché una rana può muovere solo in una posizione vuota, è facile controllare la validità della mossa (vedi linea 200).

Il resto del programma è molto chiaro; il valore di "cont" rappresenta il numero di mosse fatte.

ESECUZIONE DEL PROGRAMMA:

Il programma si aspetta un input di 2 cifre per definire le mosse: arrivo e partenza.

Saranno accettati, come prima mossa valida, soltanto i seguenti inputs:

35 45 67 75

Buon divertimento.

SALTARANA

```
100 LET a$="{SG8}▲{SG8}▲{SG8}▲{SG8}▲▲▲{SG6}▲{SG6}▲
    {SG6}▲{SG6}"
110 LET b$=a$
120 LET cont=0
130 CLS
140 PRINT AT 5,3;b$;AT 6,3;"1 2 3 4 5 6 7 8 9"
150 IF b$(1 TO 7)=a$(11 TO 17) AND b$(11 TO 17)=a$(1
    TO 7) THEN GO TO 250
160 INPUT "Inserisci la tua mossa ";k$
170 IF LEN k$<>2 THEN GO TO 160
180 LET da=2*(CODE k$(1)-48)-1
190 LET a=2*(CODE k$(2)-48)-1
200 IF b$(a)<>"▲" OR ABS (a-da)>4 THEN GO TO 160
210 LET cont=cont+1
220 LET b$(a)=b$(da)
230 LET b$(da)=" "
240 GO TO 140
250 PRINT "Lo hai fatto in ";cont;" mosse"
260 INPUT "Un'altra partita? ";k$
270 IF k$="s" THEN RUN
```

Ordina i numeri

Copyright (c) Beam Software

Questo gioco prova la potenza della vostra logica! Iniziate con 9 numeri, da 1 a 9 in ordine casuale. Scopo del gioco è di ordinarli nel minor numero di mosse possibili. L'unico modo per cambiare l'ordine dei numeri sta nella possibilità di invertire l'ordine di qualche elemento. Per esempio, avendo come primi tre numeri "3 2 1", invertendo i primi tre numeri otterrete "1 2 3". Avendo invertito solo i primi due, avreste ottenuto "3 1 2". Notate che è possibile specificare solo quanti numeri si vogliono invertire. Questo programma è piuttosto breve, così sarà semplice inserirlo nel computer, e questi sfiderà certamente la vostra potenza logica.

NOTE AL PROGRAMMA:

Questo programma illustra l'uso degli arrays, e mostra anche come si può rendere facilmente leggibile un programma se ci si prende la briga di definire variabili con nomi che abbiano un significato.

Guardate il listato — non c'è bisogno di una struttura di programma separata, poiché tutte le informazioni di cui avete bisogno sono nel programma stesso.

Notate anche che è possibile porre spazi all'inizio delle linee per rendere i cicli FOR-NEXT e altro, più evidenti.

L'uso di nomi di variabili per i GOTO e i GOSUB è un altro modo molto efficiente per seguire lo svolgimento del programma.

Questo metodo, inoltre, minimizza il lavoro che deve essere fatto qualora venisse usata la routine di RENUMBER fornita più avanti in questo libro.

USO DEGLI ARRAYS

L'uso degli arrays vi consente di riferirsi ad un particolare elemento senza dover avere un nome separato per ogni elemento.

All'inizio del programma viene dimensionato un array, ogni elemento del quale viene messo a zero. Notate che la dimensione dell'array è definita in una variabile, così se volete provare la vostra logica con una lunga lista di numeri, vi basta semplicemente cambiare quell'unica linea.

Notate il modo in cui i numeri vengono fatti apparire in modo casuale — viene preso ogni numero a turno e sostituito al numero che si trova in una posizione scelta a caso. Questo algoritmo è un modo molto utile ed efficiente di mischiare le carte in un programma che gioca a carte.

ORDINA I NUMERI

```
100 LET print=600
110 LET inverti=400
120 LET input=300
130 LET dimens=9
140 DIM a(dimens)
150 FOR c=1 TO dimens
160   LET a(c)=c
170 NEXT c
180 FOR c=dimens TO 2 STEP -1
190   LET invers=1+INT (dimens*RND)
200   LET temp=a(c)
210   LET a(c)=a(invers)
220   LET a(invers)=temp
230 NEXT c
240 LET cont=0
250 GO SUB print
300 INPUT "Quanti numeri vuoi invertire? ";quanti

310 IF quanti=0 THEN STOP
320 IF quanti<=dimens THEN GO TO inverti
330 PRINT dimens;" e' il massimo che puo' essere
   invertito"
340 GO TO input
400 PRINT "Inversione di ";quanti;" numeri"
410 LET cont=cont+1
420 FOR d=1 TO INT (quanti/2)
430 LET temporaneo=a(d)
440 LET specchio=quanti-d+1
450 LET a(d)=a(specchio)
460 LET a(specchio)=temporaneo
470 NEXT d
480 GO SUB print
490 FOR d=1 TO dimens
500 IF a(d)<>d THEN GO TO input
510 NEXT d
520 PRINT "Lo hai fatto in ";cont;" mosse."
530 STOP
600 POKE 23692,255
610 FOR d=1 TO dimens
620 PRINT a(d);"^";
630 NEXT d
640 PRINT
650 RETURN
```

Asteroidi nello spazio

Copyright (c) by Neil Streeter

State viaggiando nello spazio sulla vostra astronave, quando improvvisamente incontrate uno stormo spaziale.

Potete manovrare la vostra nave attraverso i detriti solamente usando i controlli del timone (tasto "5" per andare a sinistra e "8" per andare a destra) — la guida nell'iperspazio è stata disabilitata da una delle meteore. Come se ciò non bastasse, se sopravvivete allo stormo di meteore, vi accorgete che siete talmente disorientato da viaggiare nella direzione sbagliata sulla strada spaziale. Tutto il traffico spaziale sta venendo direttamente verso di voi. Dovete manovrare attraverso il traffico esattamente come facevate prima, tranne che le astronavi sono più grandi dei detriti, il che rende più difficile la cosa. Alla fine, vi schianterete: Quando succederà, leggerete il tasso di sopravvivenza sullo schermo della console dell'astronave.

STRUTTURA DEL PROGRAMMA

Questo programma simula l'uso della funzione SCROLL, disponibile sulla maggior parte di microcomputers ma non sullo Spectrum. Avete sicuramente già familiarità con l'istruzione SCROLL che compare, per esempio, quando viene listato un programma, ma forse avete notato che non esiste una istruzione direttamente disponibile per ottenere lo stesso risultato. Tale risultato può tuttavia essere ottenuto cambiando il valore della variabile SCR-CT (si fa ciò eseguendo una POKE all'indirizzo 23962 — vedi il Monitor Codice Macchina per ulteriori informazioni riguardo a PEEK e POKE), e quindi stampare due spazi alla fine dello schermo (cioè a 21,31). Quando il video subirà uno SCROLL, la vostra nave si muoverà insieme al resto, su di essa verrà stampato uno spazio, quindi verrà stampata di nuovo nella posizione corretta. In questo modo la nave rimane sulla stessa linea dello schermo mentre tutto il resto si muove. I detriti degli asteroidi e le navi spaziali appaiono a caso in fondo allo schermo, e le informazioni sulla loro posizione vengono contenute nelle variabili a, b, c, d ed e con e contenente quella della più vicina. Confrontando e con la posizione della vostra nave, il programma determina se state per schiantarvi. Lo schermo subisce due SCROLL ad ogni ciclo, così che ci sono solo 11 asteroidi alla volta. Potete eseguire uno SCROLL ad ogni ciclo, ma ciò significa che ci sarebbero troppi asteroidi sullo schermo, e dovrete tener conto del doppio di variabili.

MIGLIORAMENTI AL PROGRAMMA

Il programma è stato deliberatamente mantenuto semplice per permettervi di migliorarlo.

Il primo passo che può essere compiuto sta nell'usare le possibilità che vi vengono offerte dai caratteri definibili dall'utente per dare alla vostra astronave e ai detriti

degli asteroidi una forma più realistica.

In secondo luogo, noterete che ci sono due punti nei quali viene generato del suono attraverso l'istruzione BEEP. Lo scopo di questa istruzione non è puramente quello di creare del suono!

Provate a cancellare queste linee e sono sicuro che noterete che il programma diventa troppo veloce. Potete comunque accelerare o rallentare il programma cambiando la lunghezza delle note o cambiando il numero delle note che devono essere suonate.

C'è inoltre tempo sufficiente in ogni ciclo per aggiungere altre astronavi che vi attaccano, aggiungere la possibilità di sparare gli asteroidi e così via.

ASTEROIDI NELLO SPAZIO

```
100 LET a$="*"
110 LET n=0
120 LET a=0
130 LET b=0
140 LET c=0
150 LET d=0
160 LET t=1
170 LET x=12
180 LET r=INT (27*RND)
190 PRINT AT 21,r; INK (4*RND);a$
200 POKE 23692,255
210 PRINT AT 21,31;"▲▲"
220 BEEP .3,0
230 PRINT AT 10,x-2;"▲▲▲▲▲▲"
240 PRINT AT 11,x;"{G6}{SG6}"
250 PRINT AT 21,31;"▲▲"
260 LET n=n+t
270 IF n=100 THEN LET a$="{SG6}{SG7}"
280 IF n=104 THEN LET t=2
290 LET e=d
300 LET d=c
310 LET c=b
320 LET b=a
330 LET a=r
340 PRINT AT 10,x-2;"▲▲▲▲▲▲"
350 PRINT AT 11,x;"{G6}{SG6}"
360 IF x>=e-2 AND x<=e+t THEN GO TO 410
370 BEEP .2,5
380 IF INKEY$="5" THEN LET x=x-t
390 IF INKEY$="8" THEN LET x=x+t
400 GO TO 180
410 PRINT AT 11,x-1;"CRASH"
420 PRINT AT 0,0;"PUNTEGGIO=";n
430 BEEP 2,-12
440 PRINT AT 21,0;"Premi un tasto per riprovare"
450 IF INKEY$="" THEN GO TO 450
460 RUN
```


Orologio Spectrum

Copyright (c) Beam Software

DESCRIZIONE

Questo programma è la simulazione di un OROLOGIO IN TEMPO REALE; sono implementati e vengono mostrati sullo schermo sia un orologio normale che uno digitale. Resterete stupiti dall'accuratezza dell'OROLOGIO SPECTRUM; non solo, potete anche mettere la SVEGLIA e l'OROLOGIO SPECTRUM vi avvertirà generando un suono.

COME AVVIARE IL PROGRAMMA

Battete RUN seguito da (ENTER) per far partire il programma. Dovrete inserire le ORE, i MINUTI e i SECONDI dai quali volete che il programma parta. Notate che le ORE vanno inserite su un arco di 24 ore; ciò significa che dovete battere 13 anziché 1 per l'una del pomeriggio.

Il programma controlla la validità dell'input cosicché non potete battere un numero negativo o maggiore di 60 per minuti o secondi; non sarebbero accettati.

Dovrete fare altrettanto se volete mettere la sveglia.

Un modo per usare effettivamente questa sveglia è di usarla come sveglia mattutina: Fate partire il programma la sera, mettendo la sveglia all'ora giusta, quindi spegnete il televisore per risparmiare elettricità.

Lo SPECTRUM continuerà ad eseguire il programma finché non viene raggiunta l'ora della sveglia, quindi il suono del beeper interno vi sveglierà.

STRUTTURA DEL PROGRAMMA

Il programma usa la variabile di sistema FRAME dello SPECTRUM come orologio interno. La variabile di sistema FRAME occupa tre bytes che iniziano alla locazione 23672 con il byte meno significativo prima. Ogni 20 ms, FRAME viene incrementato di 1; in altre parole, il totale del contatore diviso cinquanta darà il numero di secondi dall'inizio del conteggio.

La struttura del programma è la seguente:

INIZIALIZZAZIONE

```
INPUT ORE MINUTI e SECONDI
SE VIENE MESSA LA SVEGLIA
  INPUT ORE MINUTI e SECONDI DELLA SVEGLIA
  VISUALIZZA L'ORA DELLA SVEGLIA
```

DISEGNA IL QUADRANTE DELL'OROLOGIO
CALCOLA IL VALORE INIZIALE DEL CONTATORE DI FOTOGRAMMI
POKE NEL CONTATORE DEL VALORE CALCOLATO QUANDO VIENE PREMUTO UN TASTO
DEFINISCE LA FUNZIONE: numero di secondi dall'inizio.
DETERMINA L'ORA D'INIZIO

CICLO PRINCIPALE

H: DETERMINA LA POSIZIONE DELLA LANCETTA DELLE ORE
M: DETERMINA LA POSIZIONE DELLA LANCETTA DEI MINUTI
S: DETERMINA LA POSIZIONE DELLA LANCETTA DEI SECONDI
DISEGNA LA LANCETTA DEI SECONDI
DISEGNA LA LANCETTA DEI MINUTI
DISEGNA LA LANCETTA DELLE ORE
CALCOLA E VISUALIZZA L'OROLOGIO DIGITALE
 determina le cifre di ore minuti e secondi
 correzione quando si passa mezzogiorno
 SE È RAGGIUNTA L'ORA DELLA SVEGLIA
 LAMPEGGIA L'OROLOGIO E GENERA IL SUONO
LEGGE FRAME FINCHÉ NON È PASSATO UN SECONDO
CANCELLA LA LANCETTA DEI SECONDI
SE È LO STESSO MINUTO ALLORA VA A S:
CANCELLA LA LANCETTA DEI MINUTI
SE È LA STESSA ORA ALLORA VA A M:
CANCELLA LA LANCETTA DELLE ORE
VA A H:

CARATTERISTICHE PARTICOLARI

DEF FN t() viene usata per facilitare frequenti riferimenti alla variabile di sistema FRAME.

Le funzioni SIN e COS vengono usate per disegnare le lancette dell'orologio.

La lancetta dei SECONDI viene mossa ogni secondo; la lancetta dei MINUTI ogni minuto e la lancetta delle ORE ogni 12 minuti.

Il programma tiene conto di inserimenti ante e post meridiani.

Nel programma è inserito un controllo per verificare la validità degli input.

NOTE SPECIALI

L'OROLOGIO SPECTRUM ha un'accuratezza pari a una deviazione di 10 secondi al giorno se il computer lavora senza alcuna operazione di INPUT/OUTPUT inclu-

dendo il suono. Notate che il computer non sarà più un orologio preciso dopo aver eseguito una operazione di INPUT/OUTPUT poiché il contatore non viene incrementato durante questo periodo.

Dopo aver raggiunto l'ora della sveglia, il computer dovrà essere resettato, cioè se volete usare ancora l'orologio dovrete dare ancora il RUN.

OROLOGIO SPECTRUM

```
100 GO SUB 360
110 LET c=PI/30
120 DEF FN t()=INT ((65536*PEEK 23674+256*PEEK 23
    673+PEEK 23672)/50): REM n. di secondi dall'i
    nizio
130 REM L'orologio parte
140 LET t1=FN t()
150 LET hi=INT (t1/720)
160 LET h=hi*c
170 LET hx=50*SIN h: LET hy=50*COS h
180 LET mi=INT (t1/60)
190 LET m=mi*c
200 LET mx=60*SIN m: LET my=60*COS m
210 LET a=t1*c: REM a e' l'angolo della lancetta
    dei secondi in radianti
220 LET sx=72*SIN a: LET sy=72*COS a
230 PLOT 131,91: DRAW sx,sy: REM Disegna la lance
    tta dei secondi
240 PLOT 131,91: DRAW mx,my
250 PLOT 131,91: DRAW hx,hy
260 GO SUB 740
270 LET t=FN t()
280 IF t<=t1 THEN GO TO 270: REM Attende un seco
    ndo
290 LET t1=t
300 PLOT 131,91: DRAW OVER 1;sx,sy: REM Cancell
    a la lancetta dei secondi
310 IF INT (t1/60)<=mi THEN GO TO 210
320 PLOT 131,91: DRAW mx,my: PLOT 131,91: DRAW
    OVER 1;mx,my
330 IF INT (t1/720)<=hi THEN GO TO 180
340 PLOT 131,91: DRAW hx,hy: PLOT 131,91: DRAW
    OVER 1;hx,hy
350 GO TO 150
360 BORDER 7: PAPER 7: INK 0: OVER 0: FLASH 0:
    CLS : PRINT "Inserisci l'ora": PRINT "(0 - 2
    3)"
370 INPUT H: IF H<0 OR H>23 THEN GO TO 370
380 PRINT FLASH 1; INK 2;H
390 LET D=0: IF H>=12 THEN LET D=1: LET H=H-12*(
    H<>12)
```

```

400 PRINT "Quanti MINUTI?": PRINT "(0 - 59)"
410 INPUT M: IF M<0 OR M>59 THEN GO TO 410
420 PRINT FLASH 1; INK 1;M
430 PRINT "Quanti SECONDI?": PRINT "(0 -59)"
440 INPUT S: IF S<0 OR S>59 THEN GO TO 440
450 PRINT FLASH 1; INK 3;S
460 PAUSE 50: CLS
470 PRINT "Vuoi mettere la SVEGLIA?": PRINT "(s o
  r n)"
480 INPUT a$: IF a$<>"s" AND a$<>"n" THEN
  GO TO 480
490 IF a$="n" THEN LET al=0: BORDER 0: PAPER 0:
  INK 7: CLS : GO TO 630
500 LET al=1: PRINT : PRINT "Ore? ";
510 INPUT ah: IF ah<0 OR ah>24 THEN GO TO 510
520 PRINT ah: LET ad=0: IF ah>11 THEN LET ad=1:
  LET ah=ah-(ah>12)*12
530 PRINT : PRINT "Minuti? ";
540 INPUT am: IF am<0 OR am>59 THEN GO TO 540
550 PRINT am
560 PRINT : PRINT "Secondi? ";
570 INPUT as: IF as<0 OR as>59 THEN GO TO 570
580 PRINT as: PAUSE 50
590 REM Disegna il quadrante
600 PAPER 0: BORDER 0: INK 7: CLS : PRINT AT 20
  ,0;"SVEGLIA": PRINT ah;" ":";am;" ":";as;" .";
610 IF ad=0 THEN PRINT "AM": GO TO 630
620 IF ad=1 THEN PRINT "PM"
630 FOR n=1 TO 12
640 PRINT AT 10-10*COS (n/6*PI),16+10*SIN (n/6*
  PI);n
650 NEXT n
660 LET dh=H: LET dm=M: LET ds=S
670 LET T=(H*3600+M*60+S)*50
680 LET b1=T-INT (T/256)*256
690 LET b2=INT (T/256)-INT (T/256↑2)*256
700 LET b3=INT (T/256↑2)-INT (T/256↑3)*256
710 IF INKEY$="" THEN GO TO 710
720 POKE 23674,b3: POKE 23673,b2: POKE 23672,b1
730 RETURN
740 LET ds=t1-INT (t1/60)*60: LET dm=INT (t1/60)-
  INT (t1/3600)*60: LET dh=INT (t1/3600)
750 IF ds=0 THEN PRINT AT 0,0;"....."
760 IF dh>=13 THEN LET dh=dh-12
770 IF dh<>12 OR ds<>0 OR dm<>0 THEN .
  GO TO 810
780 IF D=0 THEN LET D=1: LET dh=12: GO TO 810
790 IF D=1 THEN LET D=0: LET dh=0
800 LET dh=dh-(dh=12 AND D=0)*12
810 PRINT AT 0,0;dh;" ":";dm;" ":"; FLASH 1;ds

```

```
820 PRINT AT 0,8;" ";
830 IF D=0 THEN PRINT "AM"
840 IF D=1 THEN PRINT "PM"
850 IF a1=0 THEN RETURN
860 IF ah<>dh OR am<>dm OR as<>ds OR ad
   <>D THEN RETURN
870 PRINT AT 0,0;dh;" ";dm;" ";ds: FOR s=22528
   TO 23231: POKE s,128+PEEK s: NEXT s
880 BEEP 0.5,27: BEEP 0.5,20: IF INKEY$="" THEN
   GO TO 880
890 RETURN
```


3-D Mazeman

Copyright (c) by Beam Software

Siete intrappolati in un labirinto e dovete uscirne! La tensione aumenta!

Vi muovete in avanti, una via d'accesso vi appare sulla sinistra, veloci! Oh no... un'altra via senza uscita!

Riuscirete a uscirne vivi?

Questa appassionante simulazione tridimensionale vi colloca proprio nel labirinto, con l'obiettivo di uscirne. Dovete solo ringraziare che non ci sono mostri qui!! Per ora!!

Potete disporre del tempo che volete, ma sarete cronometrati. A ciascuna posizione avrete una prospettiva realistica. Potete girarvi a sinistra o a destra (usando i tasti "O" e "P" rispettivamente), muovervi in avanti (usando la sbarra spaziatrice) oppure far dietro front (usando il tasto "R").

Se vi siete persi veramente, sarebbe una buona idea disegnare una mappa.

STRUTTURA DEL PROGRAMMA:

Questo programma usa le funzioni PLOT e DRAW dello SPECTRUM.

A ciascun punto del corridoio c'è l'opzione della porta che può essere aperta oppure chiusa. Questo programma usa una diversa subroutine per segnare la via d'accesso a seconda della distanza dall'osservatore e a seconda del fatto che sia aperta o meno.

Questo spiega perché ci sono così tante subroutines da 6000 a 8508.

Una tabella a parte (linee 9100-9208) definisce il particolare labirinto in cui vi trovate.

PER DEFINIRE UN LABIRINTO:

Il labirinto usato qui è disegnato su una matrice 9x9, qualcosa di simile a una scacchiera, e ogni posizione nel labirinto è rappresentata da un quadrato della scacchiera.

Ovviamente, ogni quadrato può avere pareti in ognuna delle quattro direzioni. Questo programma usa due vettori per tener conto delle pareti, un vettore per le pareti orizzontali e un vettore per le verticali.

Devono essere definiti un totale di 18 linee, 9 verticalmente, e 9 orizzontalmente. Per ciascuna linea, mettiamo uno 0 se non c'è la parete a bloccare il nostro sentiero e un 1 se c'è.

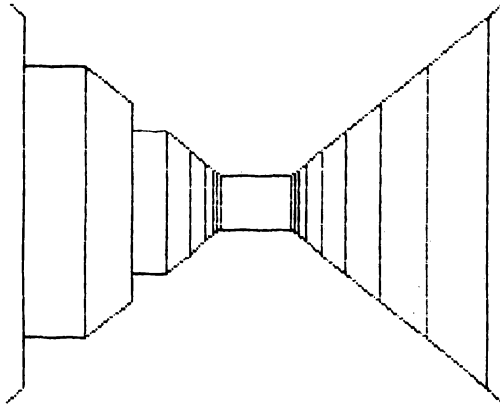
Ci sono sempre pareti sulla parte esterna del perimetro, e queste non è necessario che siano definite.

Questa informazione è memorizzata nelle istruzioni DATA 9100-9208.

Per esempio, guardando la scacchiera dall'angolo in basso a sinistra, vediamo una parete. Se fossimo in grado di superare quella parete ce ne sarebbe un'altra e un'altra ancora. Finalmente vedremo un corridoio che si estende quattro quadrati prima dell'ultima parete. Questa è l'informazione contenuta alla linea 9100.

L'uscita è definita dalla linea 156. Essa dice che se siete nella nona riga rivolto indietro, e meno di 5 file dalla fine vedrete il cartello USCITA. Il limite di 5 dalla fine è dovuto al fatto che c'è una parete nel labirinto che blocca la visuale se siete ad almeno 6 dalla fine.

Potete facilmente adattare il programma e giocare con il vostro labirinto, ma ricordatevi di cambiare le condizioni dell'uscita.



3-D MAZEMAN

```
100 DIM v(10,10): DIM h(10,10)
110 GO SUB 9000
120 LET x=1: LET y=1: LET dx=1: LET dy=0
125 LET ti=PEEK 23672+256*PEEK 23673+4096*PEEK 2
    3674
130 LET l=9: LET lx=x+(dx=-1): LET ly=y+(dy=-1)
132 LET l=l-1: LET lx=lx+dx: LET ly=ly+dy
134 IF dx<>0 AND v(lx,ly)=0 THEN GO TO 132
136 IF dy<>0 AND h(lx,ly)=0 THEN GO TO 132
140 CLS : GO SUB 6000+1
145 LET lx=lx-(dx=-1): LET ly=ly-(dy=-1)
```

```

150 FOR i=1 TO 8
155 LET lx=lx-dx: LET ly=ly-dy: GO SUB 1000:
    NEXT i
156 IF dx=-1 AND y=9 AND x<5 THEN PRINT AT 10,14;
    INK 2;"USCITA"; INK 0
158 LET a$=INKEY$: IF a$="" THEN GO TO 158
160 IF a$<>"^" THEN GO TO 190
170 IF (dx=1 AND v(x+1,y)=0) OR (dx=-1 AND v(x,y)=
    0) OR (dy=1 AND h(x,y+1)=0) OR (dy=-1 AND h(x,
    y)=0) THEN LET x=x+dx: LET y=y+dy
190 IF x=1 AND y=9 THEN GO TO 9990
200 IF a$="r" THEN LET dx=-dx: LET dy=-dy:
    GO TO 130
210 IF a$="o" THEN GO TO 300
215 IF a$<>"p" THEN GO TO 130
220 IF ABS dx=1 THEN LET dy=-dx: LET dx=0:
    GO TO 130
230 LET dx=dy: LET dy=0: GO TO 130
300 IF ABS dx=1 THEN LET dy=dx: LET dx=0: GO TO 1
    30
310 LET dx=-dy: LET dy=0: GO TO 130
1000 IF dx<>1 THEN GO TO 1100
1010 IF h(lx,ly+1)=0 THEN GO SUB 7500+i: GO TO 105
    0
1020 GO SUB 7000+i
1050 IF h(lx,ly)=0 THEN GO TO 8500+i
1060 GO TO 8000+i
1100 IF dx<>-1 THEN GO TO 1200
1110 IF h(lx,ly)=0 THEN GO SUB 7500+i: GO TO 1150
1120 GO SUB 7000+i
1150 IF h(lx,ly+1)=0 THEN GO TO 8500+i
1160 GO TO 8000+i
1200 IF dy<>-1 THEN GO TO 1300
1210 IF v(lx+1,ly)=0 THEN GO SUB 7500+i: GO TO 125
    0
1220 GO SUB 7000+i
1250 IF v(lx,ly)=0 THEN GO TO 8500+i
1260 GO TO 8000+i
1300 IF v(lx,ly)=0 THEN GO SUB 7500+i: GO TO 1350
1310 GO SUB 7000+i
1350 IF v(lx+1,ly)=0 THEN GO TO 8500+i
1360 GO TO 8000+i
6000 PLOT 110,76: DRAW 0,24: DRAW 36,0: DRAW 0,-24:
    DRAW -36,0: RETURN
6001 PLOT 108,75: DRAW 0,26: DRAW 40,0: DRAW 0,-26:
    DRAW -40,0: RETURN
6002 PLOT 106,74: DRAW 0,28: DRAW 44,0: DRAW 0,-28:
    DRAW -44,0: RETURN
6003 PLOT 102,71: DRAW 0,34: DRAW 52,0: DRAW 0,-34:
    DRAW -52,0: RETURN

```

6004 PLOT 94,65: DRAW 0,46: DRAW 68,0: DRAW 0,-46:
DRAW -68,0: RETURN
6005 PLOT 82,57: DRAW 0,62: DRAW 92,0: DRAW 0,-62:
DRAW -92,0: RETURN
6006 PLOT 64,45: DRAW 0,86: DRAW 128,0: DRAW 0,-86:
DRAW -128,0: RETURN
6007 PLOT 40,29: DRAW 0,118: DRAW 176,0: DRAW 0,-11
8: DRAW -176,0: RETURN
6008 PLOT 8,7: DRAW 0,162: DRAW 240,0: DRAW 0,-162:
DRAW -240,0: RETURN
7000 PLOT 108,75: DRAW 2,1: DRAW 0,24: DRAW -2,1:
RETURN
7001 PLOT 106,74: DRAW 2,1: DRAW 0,26: DRAW -2,1:
RETURN
7002 PLOT 102,71: DRAW 4,3: DRAW 0,28: DRAW -4,3:
RETURN
7003 PLOT 94,65: DRAW 8,6: DRAW 0,34: DRAW -8,6:
RETURN
7004 PLOT 82,57: DRAW 12,8: DRAW 0,46: DRAW -12,8:
RETURN
7005 PLOT 64,45: DRAW 18,12: DRAW 0,62: DRAW -18,12
: RETURN
7006 PLOT 40,29: DRAW 24,16: DRAW 0,86: DRAW -24,16
: RETURN
7007 PLOT 8,7: DRAW 32,22: DRAW 0,118: DRAW -32,22:
RETURN
7008 PLOT 0,1: DRAW 8,6: DRAW 0,162: DRAW -8,6:
RETURN
7500 PLOT 108,76: DRAW 2,0: DRAW 0,24: DRAW -2,0:
RETURN
7501 PLOT 106,75: DRAW 2,0: DRAW 0,26: DRAW -2,0:
RETURN
7502 PLOT 102,74: DRAW 4,0: DRAW 0,28: DRAW -4,0:
RETURN
7503 PLOT 94,71: DRAW 8,0: DRAW 0,34: DRAW -8,0:
RETURN
7504 PLOT 82,65: DRAW 12,0: DRAW 0,46: DRAW -12,0:
RETURN
7505 PLOT 64,57: DRAW 18,0: DRAW 0,62: DRAW -18,0:
RETURN
7506 PLOT 40,45: DRAW 24,0: DRAW 0,86: DRAW -24,0:
RETURN
7507 PLOT 8,29: DRAW 32,0: DRAW 0,118: DRAW -32,0:
RETURN
7508 PLOT 0,7: DRAW 8,0: DRAW 0,162: DRAW -8,0:
RETURN
8000 PLOT 148,75: DRAW -2,1: DRAW 0,24: DRAW 2,1:
RETURN
8001 PLOT 150,74: DRAW -2,1: DRAW 0,26: DRAW 2,1:
RETURN

```

8002 PLOT 154,71: DRAW -4,3: DRAW 0,28: DRAW 4,3:
      RETURN
8003 PLOT 162,65: DRAW -8,6: DRAW 0,34: DRAW 8,6:
      RETURN
8004 PLOT 174,57: DRAW -12,8: DRAW 0,46: DRAW 12,8:
      RETURN
8005 PLOT 192,45: DRAW -18,12: DRAW 0,62: DRAW 18,1
2: RETURN
8006 PLOT 216,29: DRAW -24,16: DRAW 0,86: DRAW 24,1
6: RETURN
8007 PLOT 248,7: DRAW -32,22: DRAW 0,118: DRAW 32,2
2: RETURN
8008 PLOT 255,1: DRAW -8,6: DRAW 0,162: DRAW 8,6:
      RETURN
8500 PLOT 148,76: DRAW -2,0: DRAW 0,24: DRAW 2,0:
      RETURN
8501 PLOT 150,75: DRAW -2,0: DRAW 0,26: DRAW 2,0:
      RETURN
8502 PLOT 154,74: DRAW -4,0: DRAW 0,28: DRAW 4,0:
      RETURN
8503 PLOT 162,71: DRAW -8,0: DRAW 0,34: DRAW 8,0:
      RETURN
8504 PLOT 174,65: DRAW -12,0: DRAW 0,46: DRAW 12,0:
      RETURN
8505 PLOT 192,57: DRAW -18,0: DRAW 0,62: DRAW 18,0:
      RETURN
8506 PLOT 216,45: DRAW -24,0: DRAW 0,86: DRAW 24,0 :
      RETURN
      8507 PLOT 248,29: DRAW -32,0: DRAW 0,118: DRAW 32,0
: RETURN
      8508 PLOT 255,7: DRAW -7,0: DRAW 0,162: DRAW 7,0:
      RETURN
9000 FOR i=1 TO 9: LET h(i,1)=1: LET h(i,10)=1:
      FOR j=2 TO 9: READ a: LET h(i,j)=a: NEXT j:
      NEXT i
9010 FOR i=1 TO 9: LET v(1,i)=1: LET v(10,i)=1:
      FOR j=2 TO 9: READ a: LET v(j,i)=a: NEXT j:
      NEXT i
9020 RETURN
9100 DATA 1,1,1,0,0,0,0,1
9101 DATA 0,0,1,0,0,1,0,1
9102 DATA 1,1,0,0,0,0,1,1
9103 DATA 0,0,1,1,0,0,1,0
9104 DATA 1,1,1,0,0,1,1,0
9105 DATA 1,0,0,0,0,0,0,1
9106 DATA 1,1,1,0,1,0,0,1
9107 DATA 1,1,1,0,0,1,1,0
9108 DATA 1,0,0,0,0,0,0,0
9200 DATA 0,0,0,0,0,0,0,0

```

```
9201 DATA 1,1,1,1,0,1,0,0
9202 DATA 0,0,0,0,0,0,0,1
9203 DATA 0,1,0,0,1,0,0,0
9204 DATA 1,1,0,1,1,1,1,1
9205 DATA 1,1,1,1,1,1,0,1
9206 DATA 0,0,0,0,0,0,1,0
9207 DATA 1,1,0,0,1,1,0,0
9208 DATA 0,0,0,1,0,0,0,1
9990 LET te=PEEK 23672+256*PEEK 23673+4096*PEEK 236
      74
9991 LET t=te-ti
9992 PRINT AT 0,4;"BEN FATTO !! Ne siete usciti"
9993 PRINT AT 1,4;"in ";t/50;" secondi"
9994 STOP
```

Test di geometria

Copyright (c) by Beam Software

Le particolari domande del test incluse in questo programma riguardano la geometria, ma la struttura del programma è più generale.

Questo programma può essere usato in ogni situazione in cui siano richieste serie di molteplici domande/risposte.

Un interessantissimo aspetto dello Spectrum in un test di matematica è che la risposta può essere o un numero o una formula. In altre parole alla domanda "Qual'è l'area di un cerchio di raggio 2?", sia la risposta "4*PI" che la risposta "12.75" sarebbe accettabile. (Quest'ultima risposta sebbene non sia rigorosamente corretta sarebbe accettata, perché questo programma considera valide risposte con uno scarto massimo di più o meno 0.5).

STRUTTURA DEL PROGRAMMA:

Una breve iterazione di controllo dalle linee 100-260 sceglie la domanda, controlla la risposta data con quello che si aspetta e scrive il messaggio appropriato.

Ogni domanda è contenuta all'interno di una subroutine che non solo stampa la domanda ma specifica la risposta che esso accetterà.

Ovviamente il numero e la gamma delle domande è limitato solamente dalla vostra immaginazione.

TEST DI GEOMETRIA

```
100 CLS
110 LET q=1+INT (4*RND)
120 LET m=2+INT (20*RND)
130 GO SUB q*1000
200 INPUT r: PRINT r
210 IF ABS (r-a)>.5 THEN GO TO 240
220 PRINT "SI, la risposta e' ";a
230 GO TO 250
240 PRINT "SPIACENTE, la risposta era ";a
250 INPUT "Premi <ENTER> quando sei pronto";y$
260 RUN
1000 PRINT "Qual'e' la circonferenza di un cerchio di
      raggio ";m
1010 LET a=2*PI*m
1020 RETURN
2000 PRINT "Qual'e' l'area di un cerchio di raggio ";m
```

```
2010 LET a=PI*m*m
2020 RETURN
3000 PRINT "Qual'e' la superfice di una▲▲▲▲sfera di r
      aggio ";m
3010 LET a=4*PI*m*m
3020 RETURN
4000 PRINT "Qual'e' il volume di una sfera di raggio▲
      ";m
4010 LET a=(4/3)*PI*m*m*m
4020 RETURN
```


Re e Regine

Copyright (c) by Beam Software

Questo programma usa un formato multiplo di scelta per provare la vostra conoscenza della storia inglese.

Il computer sceglie un anno dal 1066 al 1461 e vi chiede chi regnava in quell'anno particolare. Vi sarà data un'alternativa di tre nomi. Se l'anno dato è uno di quelli dove c'è stato un cambiamento, la risposta corretta è il monarca regnante all'inizio di quell'anno. Digitate la risposta — solamente il computer sa esattamente se la vostra risposta è giusta o sbagliata.

Dopo 25 domande, vi verrà dato un risultato in percentuale. La probabilità di ottenere una risposta giusta rispondendo a caso è di 1 su 3, dovrete darvi da fare per raggiungere meno del 30%!

STRUTTURA DEL PROGRAMMA:

La struttura di questo programma è abbastanza semplice, ma quello che lo rende interessante è il modo in cui l'informazione circa la monarchia è memorizzata nel programma.

Dato che c'è una sola risposta esatta per un dato anno (per il modo in cui è strutturata la domanda), questo programma sfrutta la caratteristica dello Spectrum di poter effettuare le chiamate alle subroutines con nomi di variabili e di arrivare alla linea successiva effettivamente esistente.

Guardate il listato del programma — noterete che per esempio non ci sono numeri di linee fra 1088 e 1100. Se venisse inserita l'istruzione GOSUB 1090, il computer andrebbe alla linea 1090, se non la trovasse, proverebbe 1091, e così via, finché finalmente raggiungerebbe la linea 1100. In altre parole, ogni numero dal 1089 al 1100 ritornerà col risultato

a\$ = Guglielmo 2

Questo è un inizio abbastanza buono, dato che Guglielmo 2 regnò dal 1087 al 1100, e per gli scopi di questo test, ogni anno dal 1088 al 1100 sarebbe considerato come valido.

Vediamo cosa succederebbe se inserissimo GOSUB 1088: il programma ritornerebbe indietro senza informazioni particolari in a\$ dato che troverebbe soltanto l'istruzione RETURN.

Possiamo risolvere questo problema e tener conto di questo come, per esempio, questo programma fa nella linea 530, dicendo che se non ricevessimo alcuna risposta, dobbiamo andare alla subroutine sulla linea successiva! Cioè se 1088 non vi dà alcuna risposta, provate il 1089, la quale ci darà di nuovo Guglielmo 2.

Questo significa che tutti i numeri nei limiti di questo test daranno la risposta corretta.

La stessa struttura del programma può essere usata per ogni altro test in cui siano corretti una serie di valori che non si sovrappongono in diverse situazioni.

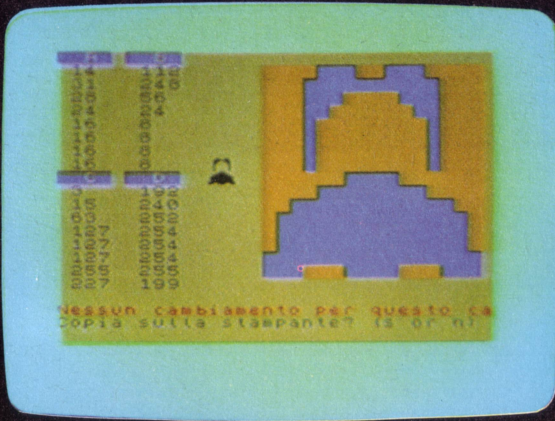
RE E REGINE D'INGHILTERRA PARTE 1:

| | | |
|-------------|-------------|--------------|
| 1066 – 1087 | Guglielmo 1 | Normanno |
| 1087 – 1100 | Guglielmo 2 | |
| 1100 – 1135 | Enrico 1 | |
| 1135 – 1154 | Stefano | |
| 1154 – 1189 | Enrico 2 | Plantageneto |
| 1189 – 1199 | Riccardo 1 | |
| 1199 – 1216 | Giovanni | |
| 1216 – 1272 | Enrico 3 | |
| 1272 – 1307 | Edoardo 1 | |
| 1307 – 1327 | Edoardo 2 | |
| 1327 – 1377 | Edoardo 3 | |
| 1377 – 1399 | Ricardo 2 | |
| 1399 – 1413 | Enrico 4 | Lancaster |
| 1413 – 1422 | Enrico 5 | |
| 1422 – 1461 | Enrico 6 | |

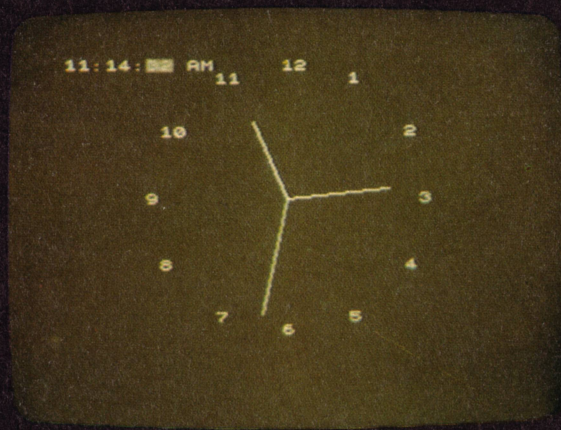
RE E REGINE D'INGHILTERRA

```
100 DIM p(3)
110 RANDOMIZE
120 LET s=0
130 FOR i=1 TO 25
140 GO SUB 500
150 LET c$=a$
160 GO SUB 500
170 LET b$=a$
180 IF b$=c$ THEN GO TO 160
190 GO SUB 500
200 IF a$=b$ OR a$=c$ THEN GO TO 190
210 PRINT i;" . Chi regnava nel "; n;" ?"
220 LET p(1)=INT (3*RND)+1
230 LET p(2)=INT (3*RND)+1
240 IF p(2)=p(1) THEN GO TO 230
250 LET p(3)=6-p(1)-p(2)
260 FOR t=1 TO 3
270 PRINT AT 5+t,5;t;" . ";
280 IF t=p(1) THEN PRINT a$
290 IF t=p(2) THEN PRINT b$
300 IF t=p(3) THEN PRINT c$
310 NEXT t
320 PRINT AT 12,0;
330 LET x$= INKEY$: IF x$<"1" OR x$>"3"
    THEN GO TO 330
340 IF VAL x$=p(1) THEN PRINT "BRAVO":
    LET s=s+1: BEEP .5,2: BEEP .2,5: BEEP .
    2,7: GO TO 370
350 PRINT "NO - era ";a$
360 FOR j=1 TO 8: BEEP RND/4,10*RND-30:
    NEXT j: BEEP 1,-25
370 PRINT FLASH 1;AT 20,0;"Premi un tasto p
    er il prossimo"
380 LET x$=INKEY$
390 IF x$="" THEN GO TO 380
400 CLS
410 NEXT i
420 PRINT "PUNTEGGIO:";4*s;" PER CENTO"
430 STOP
500 LET n=INT (396*RND)+1066
510 LET a$=" "
520 GO SUB n
530 IF a$=" " THEN GO SUB n+1
540 RETURN
1087 LET a$="Guglielmo 1"
1088 RETURN
1100 LET a$="Guglielmo 2"
1101 RETURN
```

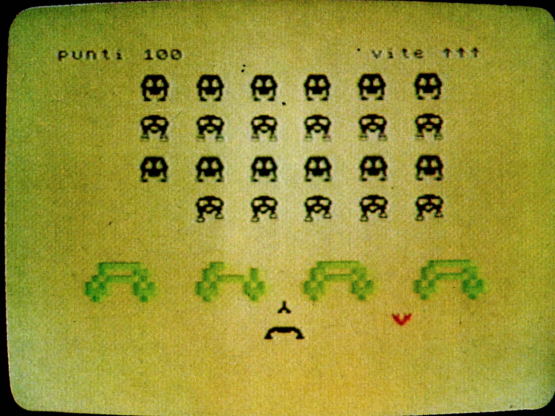
```
1135 LET a$="Enrico 1"  
1136 RETURN  
1154 LET a$="Stefano"  
1155 RETURN  
1189 LET a$="Enrico 2"  
1190 RETURN  
1199 LET a$="Riccardo 1"  
1200 RETURN  
1216 LET a$="Giovanni"  
1217 RETURN  
1272 LET a$="Enrico 3"  
1273 RETURN  
1307 LET a$="Edoardo 1"  
1308 RETURN  
1327 LET a$="Edoardo 2"  
1328 RETURN  
1377 LET a$="Edoardo 3"  
1378 RETURN  
1399 LET a$="Riccardo 2"  
1400 RETURN  
1413 LET a$="Enrico 4"  
1414 RETURN  
1422 LET a$="Enrico 5"  
1423 RETURN  
1461 LET a$="Enrico 6"  
1462 RETURN
```



Grafica ad alta
risoluzione



Orologio
Spectrum



Spectrum
Invaders

Blackjack

Copyright (c) by Beam Software

Questo è il tradizionale gioco da casinò del club ZX Spectrum. Benvenuti! Come saluto speciale a tutti i nostri ospiti, vi daremo un buono del valore di \$100 valido solamente ai nostri tavoli di blackjack.

Procediamo. Le regole delle casa sono:

- prima che le carte siano distribuite dovete decidere quanto siete disposti a scommettere. Ricordatevi che il Club ZX Spectrum non fa credito.
- il mazziere distribuisce a ciascuno una carta scoperta, poi potete scegliere quante altre carte desiderate avere.
- se non siete andati fuori gioco (cioè il vostro totale non ha superato 21) allora il mazziere distribuirà a se stesso ulteriori carte. Il mazziere chiama sempre se ha meno di 7 punti e si ferma se va oltre.

Se vincete una mano, vincete l'ammontare che avete scommesso.

Se voi e il mazziere avete lo stesso totale, vince il mazziere.

Se avete cinque carte sotto 21 vincerete a meno che il mazziere non abbia blackjack. Se avete blackjack (un totale di 21 in sole due carte) allora vincerete il doppio della posta a meno che, ancora, il mazziere non abbia anch'egli blackjack.

Incominciamo, signore e signori.

STRUTTURA DEL PROGRAMMA:

La struttura del programma è:

```
NUOVA SCOMMESSA  INPUT DELLA SCOMMESSA DEL GIOCATORE
                   SE NON RIMANGONO SOLDI, STOP
                   DÀ LA PRIMA CARTA AL MAZZIERE E AL GIOCATORE
                   PER GIOCATORE E MAZZIERE
                   SE IL GIOCATORE CHIEDE UNA CARTA
                   SE NON È COSÌ PASSA AL MAZZIERE
                   DISTRIBUISCI LA CARTA E VISUALIZZALA
                   CALCOLA IL VALORE IN MANO E VISUALIZZALO
                   SE È OLTRE IL 21 VAI A PAGAMENTO
                   SE IL MAZZIERE È OLTRE 16 VAI A PAGAMENTO
```

PAGAMENTO SE IL GIOCATORE VA FUORI GIOCO, O IL TOTALE DEL MAZZIERE È
 MAGGIORE O UGUALE AL GIOCATORE
 ALLORA VINCITA = 0
 SE C'È BLACKJACK ALLORA VINCITA RADDOPPIATA
 POOL = POOL + VINCITA
 VA ANCORA A NUOVA SCOMMESSA

La struttura di questo programma è dunque piuttosto semplice e avrete pochi problemi a modificarlo se volete cambiare le "regole della casa".

GRAFICI:

La subroutine alla linea 8000 definisce un carattere per ciascun segno. Questa subroutine ovviamente viene chiamata una volta sola all'inizio del programma.

Ogni carta viene visualizzata dalla subroutine alla linea 8500.

Viene prima disegnato il bordo della carta poi il carattere in azzurro, e poi il carattere rappresentante il seme nel colore appropriato (nero o rosso).

Adesso ci si presenta il problema di come stampare le carte. Come sapete ci sono solo 21 caratteri grafici facilmente accessibile all'uso. Noi abbiamo già usato 4 di questi per i semi, lasciandone solo 17.

Ci sono 13 carte possibili, ma usando la visualizzazione a un carattere, questo sarebbe grande solo come un carattere normale. La soluzione impiegata nel programma consiste nel guardare la forma standard di ciascun carattere memorizzato nella ROM, ampliarla 2 volte (raddoppiarne le dimensioni), e memorizzarla nel set speciale di caratteri grafici, come A B C D. Dato che non c'è un carattere standard per il "10" una versione a un carattere di esso è definita nella subroutine 8000.

Questo metodo è un po' lento, ma nel programma stesso questo è accettabile perché aumenta la tensione di sapere quale carta è stata distribuita.

Come senza dubbio sapete ogni carattere è mostrato sullo schermo dello Spectrum con un'insieme di 8 bytes, e la definizione per il carattere standard è memorizzato nella ROM a partire dalla locazione 15616. L'indirizzo iniziale di ogni carattere può essere determinato tramite la formula $15360 + 8 \star \text{codice carattere}$ (vedi linea 9020).

Il programma poi deve determinare come raddoppiare la dimensione, e memorizzarla nell'area UDG.

NOTE SPECIALI:

Noterete una strana istruzione nella linea 270

LET a = p (i) = 11

In questo programma la variabile "a" viene usata per tener conto del numero degli assi in mano. Ciò che questa linea dice è:

```
LET a = 0
IF p(i) = 11 THEN LET a = 1
```

dove p(i) è il totale delle carte per il giocatore i.

L'espressione poteva essere scritta come sopra, oppure come segue

```
LET a = (p(i) = 11)
```

poiché sappiamo che nello ZX Spectrum un'espressione sarà messa uguale a 0 se è falsa e uguale a 1 se è vera. Risulta che possiamo togliere le parentesi senza alcuna possibilità di errore, e così abbiamo l'espressione strana ma molto compatta trovata nella linea 270.

Un'altra espressione che potreste trovare strana è nella linea 530, dove viene calcolato il valore di p(i). Abbiamo appena scelto un numero di carta dal 2 al 14, rappresentato dalla variabile "c", e vogliamo sommare il valore di "c" se questo è minore o uguale a 10 (cioè una carta numerica) somma 10 se è una figura (12, 13 o 14) e sommare 1 o 11 se questo è un asso (c = 11).

Noi possiamo fare così:

```
IF c = <10 THEN...
IF c -> 11 THEN...
IF c = 11 THEN...
```

ma quella linea da sola fa quasi tutto. Dice di aggiungere il valore della carta se è minore di 12, e aggiungere 10 se il valore è maggiore di 11. Semplice, vero?

Le linee 540-560 tengono conto che l'asso sia contato come 11 o 1, semplificando enormemente il programma.

LE VOSTRE CARTE



Totale= 17

LE CARTE DELLO SPECTRUM



Totale= 19

BLACKJACK

```
100 BORDER 7: PAPER 7: INK 0: OVER 0: CLS
110 LET b$="N23456789TAJQK"
120 LET a=0: LET m=100
130 GO SUB 8000
140 DIM p(4)
150 IF m=0 THEN PRINT AT 10,1;"Vi rimangono $0";
    AT 19,1;"Spiacenti, il vostro credito e' termi
    nato. Tornate ancora quandoavrete altro denar
    o.": STOP
160 PRINT AT 10,1;"Vi rimangono $";m: PRINT "Quan
    to volete giocare?▲▲▲"
170 INPUT b
180 CLS
190 IF b>m THEN GO TO 160
200 LET m=m-b
210 PRINT INK 0;AT 0,0;"LE VOSTRE CARTE";AT 11,0
    ;"LE CARTE DELLO SPECTRUM"
220 PRINT AT 9,1;"Totale=";AT 20,1;"Totale="
230 FOR i=2 TO 1 STEP -1
240 GO SUB 500
250 NEXT i
260 FOR i=1 TO 2
270 LET a=p(i)=11
280 IF i=1 THEN INPUT FLASH 1; PAPER 2; INK 7;"
    Volete un'altra carta? ";a$: LET a$=a$+" ":
    IF a$(1)="n" THEN GO TO 340
290 GO SUB 500
300 IF p(i)>21 THEN LET i=2: GO TO 340
310 IF p(i+2)=5 THEN GO TO 340
320 IF i=2 AND p(i)>16 THEN GO TO 340
330 GO TO 280
340 NEXT i
350 IF p(1)<=21 AND p(3)=5 AND (p(2)<>21 OR p(4
    )<>2) THEN GO TO 380
360 IF p(1)=21 AND p(3)=2 AND (p(2)<>21 OR p(4)
    <>2) THEN LET b=b*1.5: PRINT FLASH 1;
    AT 5,18;"BLACKJACK": GO TO 380
370 IF (p(2)<=21 AND p(2)>=p(1)) OR p(1)>21
    THEN LET b=0
380 LET m=m+2*b
390 GO TO 130
500 LET p(i+2)=p(i+2)+1
510 LET c=INT (13*RND)+2
520 IF c=11 THEN LET a=a+1
530 LET p(i)=p(i)+c*(c<12)+10*(c>11)
540 IF p(i)<22 OR a=0 THEN GO TO 570
550 LET a=a-1
```

```

560 LET p(i)=p(i)-10
570 GO SUB 8500
580 GO SUB 9000
590 PRINT INK 0;AT 11*i-2,9;p(i)
600 RETURN
8000 DATA 0,78,209,81,81,81,78,0
8010 DATA 102,255,255,255,126,60,24,24
8020 DATA 24,60,126,255,255,126,60,24
8030 DATA 24,60,126,255,255,219,24,60
8040 DATA 60,60,219,231,231,219,24,24
8050 RESTORE 8000
8060 FOR k=USR "e" TO USR "e"+39: READ x: POKE k,x
      : NEXT k
8070 RETURN
8500 LET p=11*i-7: LET q=6*p(i+2)-5
8510 INK 5: PRINT AT p-3,q;"{G5}{4G3}G(S)5": FOR k
      =1 TO 5: PRINT AT p-3+k,q;"{G5}▲▲▲G(S)5":
      NEXT k: PRINT AT p+3,q;"{G1}{4G3}{G2}":
      INK 0
8520 LET suit=1+INT (4*RND): INK 2*(suit<3):
      PRINT AT p-2,q+3;CHR$ (148+suit)
8530 RETURN
9000 LET x=USR "a"
9010 FOR k=0 TO 31: POKE x+k,0: NEXT k
9020 LET s=8*CODE b$(c)+15360: IF b$(c)="T" THEN
      LET s=USR "e"
9030 FOR j=0 TO 7: LET v=PEEK (s+j): LET c=0:
      LET d=256
9040 FOR k=8 TO 1 STEP -1: LET d=d/2: IF v>=d
      THEN LET c=c+3*d*d: LET v=v-d
9050 NEXT k
9060 LET h=INT (c/256): LET l=c-256*h
9070 POKE x,h: POKE x+1,h: POKE x+8,l: POKE x+9,l
9080 LET x=x+2+8*(j=3)
9090 PRINT AT p,q+2;"AB";AT p+1,q+2;"CD"
9100 NEXT j
9110 RETURN

```


Fruit Machine

Copyright (c) by Beam Software

Questo programma simula la "fruit machine" che si può trovare nei bar e nei casinò.

Il principale vantaggio del gioco d'azzardo qui è che, naturalmente, se dovete perdere le 10 sterline d'ingresso non sarebbe molto importante ma può diventare anche uno svantaggio, perché non potrete incassare le vincite.

Il programma di fruit machine estenderà le nozioni che avete sull'applicazione dei caratteri definibile dall'utente. In questo programma, determineremo i 60 caratteri definibili dall'utente per la versione 16K e altri 40 per quella con più memoria. Infatti su una macchina con 48K possono essere definiti molti più caratteri grafici senza che siano richiesti altri programmi aggiuntivi.

Incominciamo signore e signori, e buona fortuna! All'inizio vi stupirete dei grafici di questo programma.

Ovviamente si vince se tutte e tre le finestre mostrano la stessa figura, con diversi importi a seconda della figura. Si vince anche se si hanno 2 finestre adiacenti con la stessa figura, eccetto se si tratta di due limoni.

CONSIDERAZIONI SUL PROGRAMMA

Il progetto di questo programma è stato realizzato per una fruit machine con tre cilindri, e ognuno di essi ruota indipendentemente. Ogni finestra contiene 20 caratteri (4x5) e mostra una figura con simboli differenti come limoni, campane, ciliege ecc.

Dato che sono ammessi solo 21 caratteri grafici, questo limita immediatamente quello che possiamo fare. Fortunatamente per noi, non è necessario avere tutti i 60 (o 100 nel caso) caratteri definiti allo stesso momento.

Una volta che un carattere grafico sia stato scritto sullo schermo, le locazioni di memoria corrispondenti a quella posizione dello schermo manterranno quella forma, senza perderla fino a che non vi si scriva altro. Quello che fate con il carattere grafico che avete definito non è importante.

Per esempio, poniamo di definire A come BIN 01010101, BIN 10101010, BIN 01010101, ecc, e che si dica

```
PINT AT 0,0; "A"
```

allora le otto locazioni di memoria corrispondenti a (0,0) ricorderanno questo come BIN 01010101, BIN 10101010 ecc, e non come "A". Possiamo cambiare la definizione A e non avremo effetti sullo schermo, e questo non accade in tutti i computer.

Questo programma perciò disegna forme, definisce i caratteri, disegna di nuovo e così via.

STRUTTURA DEL PROGRAMMA:

| <u>LINEE</u> | <u>DESCRIZIONE</u> |
|--------------|--|
| 100-200 | Definisce le variabili |
| 950-1810 | Disegna la fruit machine e controlla che il giocatore sia solvente |
| 2000-2230 | Ciclo principale Per ogni finestra Per ognuna dalle 3 alle 6 visualizzazioni Scegli una figura Per ognuna delle 5 righe Ridefinisci il set di caratteri stabilito Stampa ogni blocco Prossima riga Prossima visualizzazione Ridefinisci il set di caratteri Prossima finestra |
| 2500-2560 | Calcolare le vincite se ce ne sono. |

Il programma ottiene tutte le sue informazioni dalle istruzioni DATA, incluso i colori dell'inchiostro e della carta da usare nel disegno del blocco seguente.

PROGRAMMA ADATTO AL 16K:

Il programma così com'è non gira su un 16K. Le istruzioni DATA non sono un modo particolarmente efficiente per memorizzare informazioni (vedi note in METEOR STORM), e non sono usati accorgimenti per minimizzare l'uso della memoria.

La versione qui inclusa ha 4 diverse figure possibili: ciliege, limoni, campane e ananas. Inoltre, all'inizio di ogni gioco, tutte le finestre vengono ridefinite per mostrare un grande punto di domanda.

Il programma può essere facilmente adattato a girare in un 16K con i seguenti cambiamenti:

Sono possibili solamente 3 figure: ciliege, limoni e campane.
All'inizio di ogni tiro della leva, ogni finestra mostrerà una ciliegia.

Per effettuare queste modifiche, sono richiesti i seguenti passi:

- Cancellare le linee 8000-8053 (definiscono punto di domanda)
- Cancellare le linee 8400-8453 (definiscono gli ananas)
- Cancellare le linee 9010-9400 (definiscono l'inchiostro e la carta per il punto di domanda e l'ananas)
- Modificare così la linea 2040

LET disegno = 1 + INT (3 * RND)

Questo limita la scelta a tre figure

- Modificare così le linee 5510, 5520, 5530

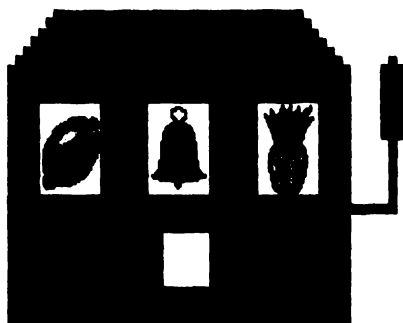
LET modello = 1

LET disegno = 1

LET vecchio dis = 1

Questo definisce le ciliege invece dei punti di domanda.

Usando alcune delle tecniche descritte nel libro, dovrebbe essere possibile adattare tutto il programma a un 16K. (Consideratelo un esercizio di programmazione!)



Ti rimangono 29
Premi <ENTER> per giocare

FRUIT MACHINE

```

100 LET cargraf=5300
110 LET disblocco=5100
120 LET primo=5
130 LET ridefgraf=5500
140 LET canccaselle=5700
150 LET soldi=10
160 DIM r(3)
170 DIM s(4)
180 RESTORE 7000
190 FOR k=1 TO 4: READ s(k): NEXT k
200 INK 7: PAPER 5: BORDER 5: CLS
950 REM disegna fruit machine
980 LET c$="{16SG8}"
1000 PRINT AT 1,5;"{G4}";c$;"{SG7}"
1010 PRINT AT 2,4;"{G4}{SG8}";c$;"{SG8}{SG7}"
1020 PRINT AT 3,3;"{G4}{2SG8}";c$;"{2SG8}{SG7}▲▲
    {G4}"
1030 PRINT AT 4,3;"{3SG8}";c$;"{3SG8}▲▲{SG8}{SG5}"
1040 PRINT AT 5,3;"{3SG8}";c$;"{3SG8}▲▲{SG8}{SG5}"
1050 PRINT AT 6,3;"{2SG8}▲▲▲▲{3SG8}▲▲▲▲{3SG8}▲▲▲▲
    {2SG8}▲▲{SG8}{SG5}"
1060 PRINT AT 7,3;"{2SG8}      {3SG8}      {3SG8}
    {2SG8}  {SG8}{SG5}"

```



```

1070 PRINT AT 8,3;"{2SG8}      {3SG8}      {3SG8}
      {2SG8} {G5}"
1080 PRINT AT 9,3;"{2SG8}      {3SG8}      {3SG8}
      {2SG8} {G5}"
1090 PRINT AT 10,3;"{2SG8}      {3SG8}      {3SG8}
      {2SG8} {G5}"
1100 PRINT AT 11,3;"{3SG8}";c$;"{3SG8}{2SG3}{SG2}"

1110 PRINT AT 12,3;"{3SG8}";c$;"{3SG8}"
1120 PRINT AT 13,3;"{7SG8}-{8SG8}-{7SG8}"
1130 PRINT AT 14,3;"{7SG8}-{8SG8}-{7SG8}"
1140 PRINT AT 15,3;"{6SG8}.{10SG8}-{6SG8}"
1150 PRINT AT 16,3;"{22SG8}"
1160 PRINT AT 17,3;"{3SG8}";c$;"{3SG8}"
1500 GO SUB canccaselle
1750 IF soldi>0 THEN GO TO 1780
1760 PRINT INK 0;AT 20,0;"Spiacente, non hai piu'
      soldi"
1770 STOP
1780 PRINT INK 0;AT 20,0;"Ti rimangono £";soldi;
      AT 21,0;"Premi <ENTER> per giocare"
1790 INPUT k$
1800 PRINT AT 20,0;"
                                     "

1810 LET soldi=soldi-.5
1820 PRINT INK 0; PAPER 0;AT 13,13;"▲▲▲";AT 14,13
      ;"▲▲▲";AT 15,13;"▲▲▲"
1830 GO SUB canccaselle
2000 REM visualizza le caselle
2010 FOR w=1 TO 3
2020 FOR d=1 TO 3+3*RND
2030 LET vecchiodis=disegno
2040 LET disegno=1+INT (4*RND)
2050 IF disegno=vecchiodis THEN GO TO 2040
2060 FOR l=5 TO 1 STEP -1
2070 GO SUB cargraf
2080 LET p=0
2090 LET modello=disegno
2100 FOR b=1 TO 5
2110 LET p=p+1
2120 GO SUB disblocco
2130 NEXT b
2140 LET modello=vecchiodis
2150 FOR b=1 TO 1-1
2160 LET p=p+1
2170 GO SUB disblocco
2180 NEXT b
2190 NEXT l
2200 NEXT d
2210 LET r(w)=disegno

```

```

2220 GO SUB ridefgraf
2230 NEXT w
2500 IF r(1)<>r(2) AND r(2)<>r(3) THEN
    GO TO 1750
2510 IF r(2)=2 THEN GO TO 1750
2520 LET vince=1
2530 IF r(1)=r(2) AND r(2)=r(3) THEN LET vince=s(
    r(1))
2540 PRINT PAPER 0;AT 13,13;"HAI";AT 14,13;"VINTO
    ";AT 15,13;"£";vince
2550 LET soldi=soldi+vince
2560 GO TO 1750
5100 REM ottiene i colori          corretti per i
    blocchi
5110 RESTORE 9000+100*modello
5120 FOR c=1 TO b-1
5130 READ ink,paper
5140 NEXT c
5150 READ ink,paper
5200 REM stampa un blocco
5210 PRINT AT primo+p,7*w-2;
5220 FOR c=1 TO 4
5230 PRINT INK ink; PAPER paper;CHR$(144+4*b-5+c
    );
5240 NEXT c
5250 RETURN
5300 REM ridefinisce i caratteri ad alta risoluzio
    ne
5310 RESTORE 8000+100*disegno+10*1
5320 FOR h=USR "a"+32*1-32 TO USR "a"+32*1-1
5330 READ byte: POKE h,byte
5340 NEXT h
5350 RETURN
5500 REM definisce i caratteri grafici per "?"
5510 LET modello=0
5520 LET disegno=0
5530 LET vecchiodis=0
5540 FOR l=5 TO 1 STEP -1
5550 GO SUB cargraf
5590 NEXT l
5600 RETURN
5700 REM riempie tutte le          caselle con ?
5710 LET mano=1: GO SUB ridefgraf: LET mano=0
5720 FOR w=1 TO 3
5730 LET d=1
5740 FOR b=1 TO 5
5750 LET p=b
5760 GO SUB disblocco
5770 NEXT b
5810 NEXT w

```

```

5820 RETURN
7000 DATA 15,3,12,7
8000 REM Dati per i caratteri grafici in alta riso
      luzione - il primo e' il punto di domanda
8010 DATA 0,0,0,0,0,0,0,0
8011 DATA 0,0,0,3,15,60,112,112
8012 DATA 0,0,0,248,252,62,14,7
8013 DATA 0,0,0,0,0,0,0,0
8020 DATA 0,0,0,0,0,0,0,0
8021 DATA 224,224,224,240,248,248,240,96
8022 DATA 3,3,3,3,3,7,6,14
8023 DATA 0,128,128,128,128,0,0,0
8030 DATA 0,0,0,0,0,0,0,0
8031 DATA 0,0,0,0,1,1,3,3
8032 DATA 28,56,112,224,192,128,0,0
8033 DATA 0,0,0,0,0,0,0,0
8040 DATA 0,0,0,0,0,0,0,0
8041 DATA 7,7,7,3,3,1,0,0
8042 DATA 0,0,128,192,192,128,0,0
8043 DATA 0,0,0,0,0,0,0,0
8050 DATA 0,0,0,0,0,0,0,0
8051 DATA 3,15,15,15,3,0,0,0
8052 DATA 0,192,192,192,0,0,0,0
8053 DATA 0,0,0,0,0,0,0,0
8100 REM Cigliegia
8110 DATA 0,0,0,0,0,0,0,0
8111 DATA 0,0,0,0,0,0,0,0
8112 DATA 0,6,6,14,30,30,62,126
8113 DATA 0,0,0,0,0,0,0,0
8120 DATA 0,0,0,0,0,0,0,0
8121 DATA 0,0,1,1,1,2,2,2
8122 DATA 127,191,31,3,0,0,0,0
8123 DATA 0,0,0,128,128,0,0,0
8130 DATA 0,0,0,0,0,0,0,0
8131 DATA 4,4,4,4,4,4,2,2
8132 DATA 0,0,0,0,0,0,0,0
8133 DATA 0,0,0,0,0,0,0,0
8140 DATA 0,0,0,0,1,1,1,1
8141 DATA 29,60,126,255,255,255,255,255
8142 DATA 224,240,120,124,252,254,254,254
8143 DATA 0,0,0,0,0,0,0,0
8150 DATA 1,1,0,0,0,0,0,0
8151 DATA 255,255,255,127,63,31,7,0
8152 DATA 254,254,252,252,248,224,128,0
8153 DATA 0,0,0,0,0,0,0,0
8200 REM Limone
8210 DATA 0,0,0,0,0,0,0,0
8211 DATA 0,0,0,0,0,0,0,1
8212 DATA 0,0,0,0,0,0,127,255
8213 DATA 0,0,0,0,0,0,16,188

```

8220 DATA 0,0,0,0,1,1,3,7
8221 DATA 7,31,127,255,254,248,240,224
8222 DATA 255,255,255,255,15,15,31,63
8223 DATA 254,254,126,60,196,246,254,254
8230 DATA 7,15,31,31,63,63,63,63
8231 DATA 192,193,199,255,255,255,255,255
8232 DATA 127,255,255,255,255,255,255,255
8233 DATA 254,254,254,254,254,252,252,252
8240 DATA 63,63,63,63,63,63,31,31
8241 DATA 255,255,255,255,255,255,255,255
8242 DATA 255,255,255,255,255,255,255,255
8243 DATA 248,248,240,224,224,192,128,0
8250 DATA 63,63,63,63,31,0,0,0
8251 DATA 255,255,255,255,0,0,0,0
8252 DATA 254,248,224,128,0,0,0,0
8253 DATA 0,0,0,0,0,0,0,0
8300 REM Campana
8310 DATA 0,0,0,0,0,0,0,0
8311 DATA 0,0,3,14,8,8,12,6
8312 DATA 0,0,192,112,16,16,48,96
8313 DATA 0,0,0,0,0,0,0,0
8320 DATA 0,0,0,0,0,0,0,0
8321 DATA 3,15,63,127,127,255,255,255
8322 DATA 192,240,252,254,254,255,255,255
8323 DATA 0,0,0,0,0,0,0,0
8330 DATA 0,1,1,1,1,1,1,1
8331 DATA 255,255,255,255,255,255,255,255
8332 DATA 255,255,255,255,255,255,255,255
8333 DATA 0,128,128,128,128,128,128,128
8340 DATA 1,3,3,3,3,7,14,56
8341 DATA 255,255,255,255,127,127,127,255
8342 DATA 255,255,255,255,255,255,255,255
8343 DATA 128,192,192,192,192,224,240,252
8350 DATA 31,3,0,0,0,0,0,0
8351 DATA 255,255,3,3,1,0,0,0
8352 DATA 255,255,128,128,0,0,0,0
8353 DATA 252,192,0,0,0,0,0,0
8400 REM ananas
8410 DATA 0,0,0,1,1,9,13,7
8411 DATA 0,4,4,14,142,158,254,189
8412 DATA 0,64,68,196,220,221,247,245
8413 DATA 0,0,16,112,224,192,128,184
8420 DATA 7,3,3,1,1,0,0,0
8421 DATA 187,221,219,235,247,251,253,255
8422 DATA 245,250,253,251,247,239,223,223
8423 DATA 240,224,192,128,128,0,0,0
8430 DATA 0,0,0,0,1,1,3,3
8431 DATA 127,127,80,219,219,219,39,178
8432 DATA 254,254,46,111,227,29,93,125
8433 DATA 0,0,0,0,128,128,128,192

8440 DATA 3,3,3,3,3,3,1,1
8441 DATA 182,183,70,57,53,53,174,209
8442 DATA 200,215,27,187,145,183,119,191
8443 DATA 192,192,192,192,128,128,128,128
8450 DATA 1,1,0,0,0,0,0,0
8451 DATA 219,255,197,237,237,127,31,0
8452 DATA 147,187,27,218,198,252,248,0
8453 DATA 0,0,0,0,0,0,0,0
9000 REM dati per l'inchiostro e la carta di ogni
blocco
9010 DATA 0,5,0,5,0,5,0,5,0,5
9100 DATA 4,1,4,1,4,1,2,1,2,1
9200 DATA 6,4,6,4,6,4,6,4,6,4
9300 DATA 7,2,7,2,7,2,7,2,7,2
9400 DATA 4,3,4,3,6,3,6,3,6,3

Valutazione dei beni

DESCRIZIONE ARTICOLO

DESCRIZIONE: sci da fondo
LUOGO: Cortina
DATA D'ACQUIS.: 1984 VITA: 1
COSTO ORIG.: 90000

ATTUALE

VALORE DI RIVENDITA 90000
COSTO DI SOSTITUZIONE 90000

1230 95 0C 08 13 DD 25 F3 3E 02 47
1240 10 FE D3 FE EE 0F 06 A4 2D 20
1250 75 05 25 F2 D8 04 05 2F 10 FE
1260 D3 FE 3E 0D 05 37 10 FE D3 FE
1270 01 0E 3B 08 6F C3 07 05 7A 53
1280 38 0C DD 6E 08 7C AD 57 3E 01
1290 37 C3 25 05 6C 16 F4 79 0B 78

Monitor codice macchina

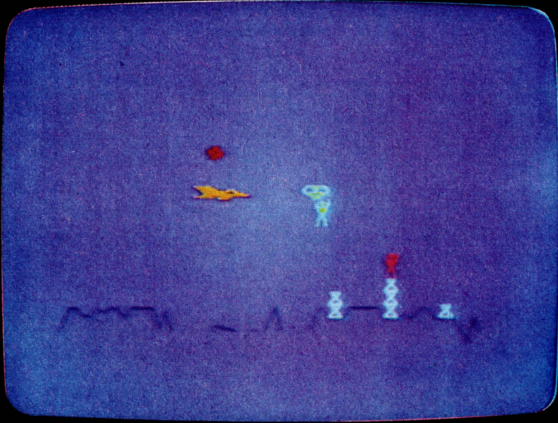
BLOCK LINE DELETE

LA CANCELLAZIONE INCLUDE
DALLA LINEA 1120
ALLA LINEA 4550

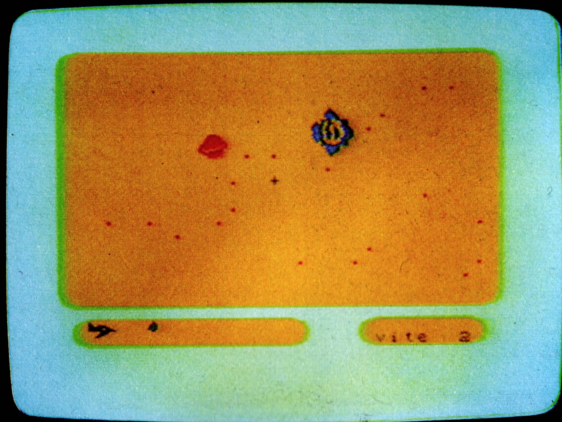
CATALE

BATTI 1120+ENTER PER CANCELLARE
IL BLOCCO

Block Line Delete



Eliminator



Meteor Storm

Bubble Sort

Copyright (c) by Beam Software

C'è gente d'ogni sorta interessata ai computers, ma nella matematica la parola 'sort' ha un diverso significato. Ci sono davvero molte differenti sorte di sort.

Un sort è un processo per il quale è possibile creare ordine dal caos, e di solito il processo di sorting viene applicato o a un gran numero di nomi o ad un vettore numerico.

Il programma proposto serve sia come dimostrazione delle operazioni coinvolte nel processo di sorting che come base per una subroutine compatta che potete usare nei vostri programmi — per esempio per ordinare una lista di nomi e indirizzi oppure per ordinare una mano di carte in una partita di bridge.

La dimostrazione di questi sort è graficamente molto divertente così come abbastanza lenta da permettere una buona comprensione del processo di sorting.

IL BUBBLE SORT:

Il primo programma presentato è un bubble sort tradizionale. Quello che fa il programma è confrontare due numeri adiacenti in una lista: se i numeri sono al posto sbagliato (in altre parole, se il numero più alto viene prima di quello più basso) allora li scambia.

Se lanciate questo programma, vi rendete subito conto del perché è detto bubble sort — i numeri più piccoli sembrano gorgogliare verso l'alto mentre i più grandi scendono verso il basso.

Il programma continuerà a scambiare numeri adiacenti, ancora e ancora, ottenendo ogni volta la lista di numeri un po' più ordinata di prima finché finalmente non dovrà più essere fatto ordine.

Il programma è molto ingegnoso nella sua esecuzione, e potete vedere quanto duramente debba lavorare per riordinare il disordine.

Vi renderete subito conto che ci deve essere un modo migliore per sistemare le cose.

BUBBLE SORT MODIFICATO:

Il listato del secondo programma è un bubble sort modificato.

Ora il programma non è pensato solamente per scambiare numeri adiacenti, ma è predisposto per scambiare numeri a distanze maggiori.

Questo elimina circa la metà delle sostituzioni da fare rispetto ad un bubble sort tradizionale.

Potete o inserire il secondo programma come listato o, se avete già inserito il programma 1, tutto ciò che bisogna fare è cambiare le linee 320, 330, 340, 500 e 510.

SUBROUTINE SORT:

A parte gli scopi dimostrativi, potete usare i sorts nei vostri programmi. Il programma 3 dà il listato del bubble sort modificato in forma di subroutine. (linee 8000-8030) che potete usare nei vostri programmi.

Questa subroutine presuppone che i numeri da ordinare siano in un vettore p(n), dove n è il numero di elementi da ordinare.

Il programma 3 può essere ottenuto dal programma 2 cancellando le linee 110-130 e le linee da 280 in avanti e quindi inserendo le nuove linee 280-300-1000-1040 e 8000-8030.

Come potete vedere da questo programma, 15 numeri possono essere ordinati in circa 3 secondi, includendo il tempo necessario per stampare i 15 numeri due volte.

```
5.1425323
6.0550842
8.676239
6.7261558
10.472961
5.4782867
6.8822632
7.1800385
11.702286
14.512955  9.4732056
6.4977417
8.3413849
11.612564
1.947113
```

VERSIONE SORT 1

```
100 BORDER 7: PAPER 7: INK 0: OVER 0: CLS
110 RESTORE 600
120 FOR i=USR "a" TO USR "d"-1
130 READ x: POKE i,x: NEXT i
140 LET n=15
150 DIM p(n)
160 PRINT "Vuoi scegliere i tuoi numeri (s o n) ?
"
170 LET b$=INKEY$: IF b$="" THEN GO TO 170
180 IF b$="n" OR b$="N" THEN GO TO 240
190 PRINT "Inserisci 15 numeri da ordinare"
200 FOR y=1 TO n
210 INPUT p(y): PRINT p(y)
220 NEXT y
230 GO TO 270
240 FOR x=1 TO n
250 LET p(x)=15*RND+1
```

```

260 NEXT x
270 CLS
280 FOR x=1 TO n
290 PRINT AT x,0;p(x)
300 NEXT x
310 LET m=18
320 FOR i=n-1 TO 1 STEP -1
330 FOR j=1 TO i
340 LET L=j+1
350 IF p(j)<=p(1) THEN GO TO 500
360 LET b$="AAAAAAAAAA"( TO LEN STR$ p(j)):
PRINT AT j,0; OVER 1; PAPER 5;b$
370 IF l=m THEN GO TO 390
380 FOR x=m TO 1 STEP SGN (1-m): PRINT INK 2;
AT x,18;"ABC";AT x+SGN (m-1),18;"AAA": NEXT x

390 LET y=LEN STR$ p(1): FOR x=17 TO y STEP -1:
PRINT INK 2;AT 1,x;"ABC.": NEXT x
400 FOR x=0 TO 10: PRINT AT 1,x;"A";p(1); INK 2;"
ABC": NEXT x
410 FOR x=1 TO j STEP -1: PRINT AT x,11;p(1);
INK 2;"ABC";AT x+1,11;"AAAAAAAAAAAAAAAA":
NEXT x
420 PRINT OVER 1;AT j,0;p(j)
430 FOR x=j TO 1: PRINT OVER 1;AT x,0;p(j);AT x,
0;p(j): NEXT x: PRINT AT 1,0;p(j)
440 FOR x=10 TO 0 STEP -1: PRINT AT j,x;p(1);
INK 2;"ABC";"A": NEXT x
450 FOR x=y TO 17: PRINT AT j,x; INK 2;"ABC":
NEXT x
460 LET m=j
470 LET t=p(1)
480 LET p(1)=p(j)
490 LET p(j)=t
500 NEXT j
510 NEXT i
520 FOR x=m TO 18: PRINT AT x,18; INK 2;"ABC";
AT x-1,18;"AAA": NEXT x
530 PRINT AT 21,0;"Vuoi ripetere ? "
540 LET b$=INKEY$: IF b$="" THEN GO TO 540
550 IF b$="s" OR b$="S" THEN RUN 100
560 IF b$="S" OR b$="S" THEN RUN 100
600 DATA 0,0,2,6,15,30,63,254

610 DATA 0,0,63,127,241,100,238,4
620 DATA 63,255,241,228,158,223,14,4

```

VERSIONE SORT 2

```

100 BORDER 7: PAPER 7: INK 0: OVER 0: CLS
110 RESTORE 600
120 FOR i=USR "a" TO USR "d"-1
130 READ x: POKE i,x: NEXT i
140 LET n=15
150 DIM p(n)
160 PRINT "Vuoi scegliere i tuoi numeri (s o n) ?
"
170 LET b$=INKEY$: IF b$="" THEN GO TO 170
180 IF b$="n" OR b$="N" THEN GO TO 240
190 PRINT "Inserisci 15 numeri da ordinare"
200 FOR y=1 TO n
210 INPUT p(y): PRINT p(y)
220 NEXT y
230 GO TO 270
240 FOR x=1 TO n
250 LET p(x)=15*RND+1
260 NEXT x
270 CLS
280 FOR x=1 TO n
290 PRINT AT x,0;p(x)
300 NEXT x
310 LET m=18
320 FOR j=1 TO n-1
330 FOR i=j+1 TO n
340 LET l=n+j-i+1
350 IF p(j)<=p(l) THEN GO TO 500
360 LET b$="^^^^^^^^^^"( TO LEN STR$ p(j)):
PRINT AT j,0; OVER 1; PAPER 5;b$
370 IF l=m THEN GO TO 390
380 FOR x=m TO 1 STEP SGN (1-m): PRINT INK 2;
AT x,18;"ABC";AT x+SGN (m-1),18;"^": NEXT x
390 LET y=LEN STR$ p(1): FOR x=17 TO y STEP -1:
PRINT INK 2;AT 1,x;"ABC^": NEXT x
400 FOR x=0 TO 10: PRINT AT 1,x;"^";p(1); INK 2;"
ABC": NEXT x
410 FOR x=1 TO j STEP -1: PRINT AT x,11;p(1);
INK 2;"ABC";AT x+1,11;"^^^^^^^^^^^^^^^^":
NEXT x
420 PRINT OVER 1;AT j,0;p(j)
430 FOR x=j TO 1: PRINT OVER 1;AT x,0;p(j);AT x,
0;p(j): NEXT x: PRINT AT 1,0;p(j)
440 FOR x=10 TO 0 STEP -1: PRINT AT j,x;p(1);
INK 2;"ABC";"^": NEXT x
450 FOR x=y TO 17: PRINT AT j,x; INK 2;"^ABC":
NEXT x

```

```

460 LET m=j
470 LET t=p(1)
480 LET p(1)=p(j)
490 LET p(j)=t
500 NEXT i
510 NEXT j
520 FOR x=m TO 18: PRINT AT x,18; INK 2;"ABC";
    AT x-1,18;"^.^": NEXT x
530 PRINT AT 21,0;"Vuoi ripetere ? "
540 LET b$=INKEY$: IF b$="" THEN GO TO 540
550 IF b$="s" OR b$="S" THEN RUN 100
560 IF b$="s" OR b$="S" THEN RUN 100
600 DATA 0,0,2,6,15,30,63,254
610 DATA 0,0,63,127,241,100,238,4
620 DATA 63,255,241,228,158,223,14,4

```

VERSIONE SORT 3

```

100 BORDER 7: PAPER 7: INK 0: OVER 0: CLS
140 LET n=15
150 DIM p(n)
160 PRINT "Vuoi scegliere i tuoi numeri (s o n) ?"

170 LET b$=INKEY$: IF b$="" THEN GO TO 170
180 IF b$="n" OR b$="N" THEN GO TO 240
190 PRINT "Inserisci 15 numeri da ordinare"
200 FOR y=1 TO n
210 INPUT p(y): PRINT p(y)
220 NEXT y
230 GO TO 270
240 FOR x=1 TO n
250 LET p(x)=15*RND+1
260 NEXT x
270 CLS
280 GO SUB 1000: GO SUB 8000
290 GO SUB 1000
300 STOP
1010 FOR x=1 TO n
1020 PRINT AT x,0;p(x)
1030 NEXT x
1040 RETURN
8000 REM Subroutine sort
8010 FOR j=1 TO n-1: FOR i=j+1 TO n: LET L=n+j-i+1
8020 IF p(1)<=p(j) THEN LET t=p(1): LET p(1)=p(j):
    LET p(j)=t
8030 NEXT i: NEXT j: RETURN

```


Equazioni simultanee

Copyright (c) by Beam Software

Questo programma risolve due equazioni simultanee del tipo:

$$ax + bx + c = 0$$

Questa è la forma generale dell'equazione della retta, e la 'soluzione' di due linee di questo tipo è il termine matematico del punto in cui le due linee si incrociano.

Vi sarà chiesto dal programma di inserire i valori A, B e C per ciascuna delle due equazioni, e il programma troverà per quali valori di x e y le linee si incontrano.

Se non c'è soluzione, — in altre parole le due linee sono parallele e non si incontrano — il programma ve lo dirà.

Il programma quindi disegnerà per voi ciascuna linea e vi mostrerà dove esse si incrociano.

STRUTTURA DEL PROGRAMMA:

La parte del programma che riguarda effettivamente il calcolo è abbastanza breve — la maggior parte del programma consiste nell'ottenere da voi informazioni corrette e nel visualizzare i risultati (incluso il grafico).

La linea 830 calcola il determinante D. Se D è zero, la soluzione non esiste.

Il blocco da 900 a 990 determina in quale quadrante si trova l'intersezione delle due linee, e in quali direzioni devono essere disegnati i grafici.

EQUAZIONI SIMULTANEE

```
100 REM EQUAZIONI
101 REM
110 REM equazioni simultanee
120 BORDER 7: INK 1: PAPER 7: FLASH 0: INVERSE 0:
    CLS
130 PRINT AT 0,5; INVERSE 1;"EQUAZIONI SIMULTANEE"

140 PRINT AT 2,0; INK 2; INVERSE 1;"a1X+b1Y+c1=0"
150 PRINT AT 2,16; PAPER 6; INK 0;"a2X+b2Y+c2=0"
160 PRINT AT 4,0;"a1=";AT 4,16;"a2=";AT 5,0;"b1=";
    AT 5,16;"b2=";AT 6,0;"c1=";AT 6,16;"c2="
170 DIM C(6): REM vettore coefficienti
180 DIM S$(7)
190 DIM X$(7)
200 DIM Y$(7)
```

```

500 REM
510 REM inserimento prima equazione
520 FOR I=1 TO 3
530 PRINT AT I+3,3; FLASH 1;">"; FLASH 0; INK 2;
    INVERSE 1;"^^^^^^^^^^^^^^^^"
540 INPUT S$: LET C(I)=VAL S$
550 PRINT AT I+3,3;"^^^^^^^^^^^^^^^^";AT I+3,4;ABS C(
    I)
560 IF SGN C(I)=-1 THEN PRINT AT I+3,3;"-"
570 NEXT I
600 REM
610 REM inserimento della seconda equazione
620 FOR I=4 TO 6
630 PRINT AT I,19; FLASH 1;">"; FLASH 0; PAPER 6;"
    ^^^^^^^^^^^^^^^^^"
640 INPUT S$: LET C(I)=VAL S$
650 PRINT AT I,19;"^^^^^^^^^^^^^^^^";AT I,20;ABS C(I)

660 IF SGN C(I)=-1 THEN PRINT AT I,19;"-"
670 NEXT I
800 REM
810 REM calcola le soluzioni
820 PRINT AT 7,0;
830 LET D=C(2)*C(4)-C(1)*C(5)
840 IF D=0 THEN PRINT INVERSE 1;"PARALLELE:";
    INVERSE 0;" "; FLASH 1;"NON C'E' SOLUZIONE":
    PAUSE 250: RUN
850 LET A=(C(3)*C(5)-C(2)*C(6))/D
860 LET X$=STR$ A: LET A=VAL X$
870 LET B=(C(1)*C(6)-C(3)*C(4))/D
880 LET Y$=STR$ B: LET B=VAL Y$
890 PRINT INVERSE 1;"SOLUZIONI"; INVERSE 0;" X= "
    ;X$;" Y= ";Y$
900 REM
910 REM disegna gli assi
920 IF A>=0 AND B>=0 THEN LET ox=72: LET oy=0:
    PRINT AT 21,8;"O";AT 8,8;"Y";AT 21,23;"X"
930 IF A>=0 AND B<0 THEN LET ox=72: LET oy=111:
    PRINT AT 8,8;"O";AT 21,8;"Y";AT 8,23;"X"
940 IF A<0 AND B>=0 THEN LET ox=183: LET oy=0:
    PRINT AT 21,23;"O";AT 21,8;"X";AT 8,23;"Y"
950 IF A<0 AND B<0 THEN LET ox=183: LET oy=111:
    PRINT AT 8,23;"O";AT 8,8;"X";AT 21,23;"Y"
960 LET dx=(SGN A>=0)-(SGN A<0)
970 LET dy=(SGN B>=0)-(SGN B<0)
980 PLOT ox,oy: DRAW INK 0;dx*111,0: PLOT ox,oy:
    DRAW INK 0;0,dy*111
990 GO SUB 1500: REM plot lines
1000 INPUT "Altre equazioni?(s o n)";k$
1100 IF k$="n" OR k$="N" THEN STOP

```



```

1110 IF k$="y" OR k$="Y" THEN RUN
1120 GO TO 1000
1500 REM
1510 LET rx=ABS (55/A): LET ry=ABS (55/B)
1520 REM draw first line
1530 PRINT AT 2,0; INK 2; OVER 1; FLASH 1;".....
      ^^^^^"
1540 IF C(1)=0 THEN LET ix1=0: LET iy1=-C(3)/C(2):
      LET ix2=2*A: LET iy2=iy1: GO TO 1620
1550 IF C(2)=0 THEN LET iy1=0: LET ix1=-C(3)/C(1):
      LET iy2=2*B: LET ix2=ix1: GO TO 1620
1560 LET ix1=2*A: LET iy1=-(C(3)+C(1)*ix1)/C(2)
1570 IF dy*iy1<0 THEN LET iy1=0: GO TO 1590
1580 IF (ABS iy1-ABS (2*B))>0 THEN LET iy1=2*B
1590 LET ix2=0: LET iy2=-C(3)/C(2)
1600 IF dy*iy2<0 THEN LET iy2=0: GO TO 1620
1610 IF (ABS iy2-ABS (2*B))>0 THEN LET iy2=2*B
1620 PLOT ox+rx*ix1,oy+ry*iy1
1630 DRAW (ix2-ix1)*rx,(iy2-iy1)*ry
1640 PAUSE 100
1650 PRINT AT 2,0; INK 2; INVERSE 1;"a1x+b1y+c1=0"
1800 REM
1810 REM draw second line
1820 PRINT AT 2,16; OVER 1; FLASH 1; PAPER 6; INK 0
      ;"....."
1830 PLOT OVER 1;ox+rx*ix1,oy+ry*iy1
1840 DRAW OVER 1;(ix2-ix1)*rx,(iy2-iy1)*ry
1850 IF C(4)=0 THEN LET ix3=0: LET iy3=-C(6)/C(5):
      LET ix4=2*A: LET iy4=iy3: GO TO 1930
1860 IF C(5)=0 THEN LET iy3=0: LET ix3=-C(6)/C(4):
      LET iy4=2*B: LET ix4=ix3: GO TO 1930
1870 LET ix3=2*A: LET iy3=-(C(6)+C(4)*ix3)/C(5)
1880 IF dy*iy3<0 THEN LET iy3=0: GO TO 1900
1890 IF (ABS iy3-ABS (2*B))>0 THEN LET iy3=2*B
1900 LET ix4=0: LET iy4=-C(6)/C(5)
1910 IF dy*iy4<0 THEN LET iy4=0: GO TO 1930
1920 IF (ABS iy4-ABS (2*B))>0 THEN LET iy4=2*B
1930 PLOT ox+ix3*rx,oy+iy3*ry
1940 DRAW (ix4-ix3)*rx,(iy4-iy3)*ry
1950 PAUSE 100
1960 PRINT AT 2,16; INK 0; PAPER 6;"a2x+b2y+c2=0"
1970 PLOT ox+ix1*rx,oy+iy1*ry
1980 DRAW (ix2-ix1)*rx,(iy2-iy1)*ry
1990 RETURN

```


Fuga spaziale

Copyright (c) Clifford Ramshaw

DESCRIZIONE

Siete stati catturati da Alien e siete arenati in una torre curata da un ROBOT di ALIEN.

Dovete combattere con la forza di gravità che vi trascinerà a terra e con il potente raggio trattore di ALIEN che vi trascina verso la torre.

Per di più, il ROBOT di ALIEN, una volta scoperto che state scappando, vi sparerà con un potente laser aspirante. Quando verrete colpiti sarete aspirati indietro e imprigionati di nuovo nella torre. BUONA FORTUNA!!

COME FAR PARTIRE IL PROGRAMMA

Per far iniziare il programma, digitate RUN (ENTER).

Usare il tasto 'w' per muoversi in alto e il tasto 'd' per muoversi avanti.

Dovete muoversi sopra il livello della cella prima di potervi muovere in avanti. A meno che non vi muoviate in avanti, sarete spinti indietro e aspirati di nuovo verso la torre. Analogamente se non premerete 'w' andrete di nuovo fino al livello della cella.

Il gioco si fermerà ogni volta che venite colpiti o che riuscite a scappare dalla prigione.

Premete un qualsiasi tasto per ricominciare il gioco.

STRUTTURA DEL PROGRAMMA

Questo programma usa speciali caratteri definibili dall'utente per disegnare il ROBOT di ALIEN e il vostro veicolo spaziale.

Viene mantenuto un contatore in tempo reale usando la variabile di sistema FRAME per registrare il tempo trascorso prima di essere ricatturato dal ROBOT o, se siete fortunati, il tempo per sfuggire.

La struttura del programma è la seguente:

INIZIALIZZAZIONE

DEFINISCE LE VARIABILI SCHERMO

DEFINISCE I CARATTERI GRAFICI DEFINIBILI DALL'UTENTE

CICLO PRINCIPALE

DISEGNA LA FORMA DELLA TORRE E LA LINEA DEL MARGINE DI SICUREZZA

INIZIALIZZA LE VARIABILI

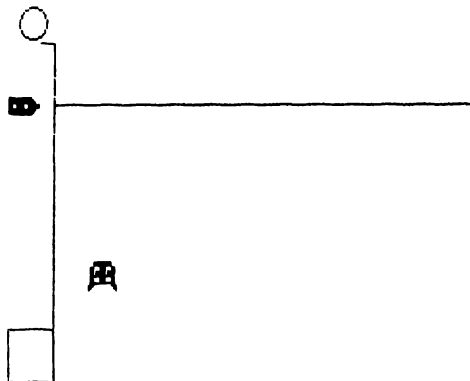
L1: MEMORIZZA LA VECCHIA POSIZIONE

CALCOLA LE NUOVE POSIZIONI
 SE NON CI SI MUOVE IN AVANTI
 genera un suono
 DISEGNA LA FORMA ALTERNATA DELL'ASTRONAVE
 SE VIENE RAGGIUNTA LA LINEA DEL MARGINE
 prepara il messaggio
 vai a T1:
 SE NON FUOCO E RND <.9
 vai a L1:
 SE FLAG FUOCO NON È ON
 metti a uno il flag fuoco
 calcola la posizione finale del ROBOT
 CANCELLA IL ROBOT VECCHIO
 MUOVI IL ROBOT VERSO LA POSIZIONE FINALE
 DISEGNA IL NUOVO ROBOT
 SE NON È STATA RAGGIUNTA LA POSIZIONE FINALE
 vai a L1:
 SPARA IL RAGGIO LASER (PLOT & PLOT OVER)
 SE L'ASTRONAVE NON È STATA COLPITA
 vai a L1:
 ESEGUI LA ROUTINE RIDISEGNA
 PREPARA IL MESSAGGIO
 T1: CALCOLA TEMPO TRASCORSO
 SCRIVI IL MESSAGGIO FINALE

NOTE SPECIALI

Il TEMPO TRASCORSO prima che il gioco sia finito con successo oppure tragicamente è influenzato dal tempo preso da ogni suono generato nel programma; ma è vicino al tempo reale.

Alla fine del gioco, la pressione di ogni tasto ricomincerà il gioco.



FUGA SPAZIALE

```
100 REM ESCAPE
110 REM SPACE ESCAPE
120 GO TO 8000
500 REM
510 REM definisce la forma del la nave
520 FOR I=USR "A" TO (USR "J"+7)
550 READ m: POKE I,m
570 NEXT I
580 DATA 0,0,0,30,30,49,61,61,0,0,0,60,60,198,246
,246,63,49,49,63,14,14,2,0,254,198,198,254,56
,56,32,0
590 DATA 0,0,0,15,15,49,49,55,0,0,0,248,248,198,1
98,222,55,63,49,49,63,112,96,64,222,254,198,1
98,254,7,3,1
600 DATA 255,255,255,207,207,255,255,255,240,248,
252,63,63,252,248,240
1000 REM
1010 REM routine che disegna la torre
1020 CIRCLE INK 0;12,167,7
1030 INK 0: PLOT 16,158: DRAW 7,0: DRAW 0,-158:
DRAW -23,0: DRAW 0,23: DRAW 23,0
1040 PLOT 255,23: DRAW 0,-23
1050 FOR I=23672 TO 23674: REM inizializza il con
tatore tempo
1060 POKE I,0
1070 NEXT I
1080 RETURN
2000 REM
2010 REM intrappolato di nuovo
2020 PLOT 24,(21-ng)*8+4: DRAW INK 2;(D-3)*8-1,0
2030 FOR I=(ID+(D>ID)) TO 0 STEP -1
2050 PRINT AT H,I;"AB": PRINT AT H+1,I;"CD."
2060 BEEP .05,(36+I)
2070 NEXT I
2090 FOR J=RH TO 18
2100 PRINT AT J,0;"AA": PRINT AT J+1,0;"AA"
2110 PRINT AT J+1,0;"AB": PRINT AT J+2,0;"CD"
2120 BEEP .05,(24+J)
2130 NEXT J
2140 RETURN
8000 REM
8010 REM ciclo principale
8020 BORDER 6: PAPER 6: INK 0: OVER 0: FLASH 0:
CLS
8030 RESTORE : GO SUB 500: REM definisce i caratt
eri grafici per la nave
8040 GO SUB 1000: REM disegna latorre
```

```

8050 LET S=30: LET H=20: LET D=3: LET sh=0: LET og
    = INT (RND*17)+2: LET fr=sh
8060 LET OH=H: LET OD=D
8070 LET k$=INKEY$
8080 LET H=H+(H<19)-(k$="w" AND H>1)*1.5: LET IH=
    INT H
8090 LET D=D+(k$="d" AND H<19)*1.5-(D>3)*.3: LET I
    D=INT D
8100 IF k$<>"d" THEN BEEP .0125,-24
8110 IF D>30 THEN LET D=30
8120 IF sh=0 THEN GO TO 8170
8130 LET sh=0
8140 PRINT AT OH,OD;"^.^": PRINT AT OH+1,OD;"^.^"
8150 PRINT AT H,D;"EF": PRINT AT H+1,D;"GH"
8160 GO TO 8200
8170 LET sh=1
8180 PRINT AT OH,OD;"^.^": PRINT AT OH+1,OD;"^.^"
8190 PRINT AT H,D;"AB": PRINT AT H+1,D;"CD"
8200 IF D=30 THEN PRINT AT 21,0;"SEI FUGGITO IN "
    ;: GO TO 8370
8210 IF fr=0 AND RND<.9 OR H>19 THEN GO TO 8060
8220 IF fr=1 THEN GO TO 8270
8230 LET ng=INT (INT (RND*2)+H)-1: LET ng=ng-(ng>1
    8)
8240 IF ng<2 THEN LET ng=2
8250 LET gm=(ng>og)-(ng<og)
8260 LET fr=1
8270 IF ng=og THEN GO TO 8310
8280 PRINT AT og,0;"^.^": LET og=og+gm
8290 PRINT AT og,0; INK 2;"IJ"
8300 GO TO 8060
8310 PLOT 24,(21-ng)*8+4: DRAW INK 2;214,0:
    BEEP .2,20
8320 PLOT OVER 1;24,(21-ng)*8+4: DRAW INK 2;
    OVER 1;214,0: BEEP .1,0
8330 LET RH=(IH+(H>IH)): IF ng<>RH AND ng<>RH+
    1 OR RH>19 THEN LET fr=0: GO TO 8060
8340 GO SUB 2000
8350 PRINT AT 21,0;"HAI RESISTITO ";
8370 LET T=INT (((65536*PEEK 23674+256*PEEK 23673+
    PEEK 23672)/(50*60))*100)/100
8380 PRINT T;" MINUTI"
8390 IF INKEY$="w" OR INKEY$="d" THEN GO TO 8390
8400 IF INKEY$="" THEN GO TO 8400
8410 CLS : GO TO 8040

```

Allunaggio

Copyright (c) Clifford Ramshaw

DESCRIZIONE

Siete il pilota di una capsula spaziale comandata a distanza da fare atterrare sulla superficie lunare.

Ogni volta che fallite l'atterraggio distruggerete la capsula e renderete anche impossibile l'atterraggio in quel punto. Dovete trovare un altro posto per l'atterraggio e rientrare con un'altra capsula.

Atterrando con successo, verrà innalzata una bandiera per segnalare il vostro diritto sul territorio.

La capsula subisce una forza costante che la muove verso destra. Allo stesso tempo, viene trascinata giù dalla gravità lunare. BUONA FORTUNA!!

COME FAR PARTIRE IL PROGRAMMA

Si inizia il programma, digitando RUN (ENTER).

Usate '5' per muovervi verso sinistra, '7' per incrementare la vostra altezza.

La capsula spaziale sarà mossa costantemente in avanti e allo stesso tempo sarà trascinata giù dalla gravità lunare.

Dovete atterrare con le due gambe della capsula spaziale esattamente sul punto d'atterraggio AZZURRO CIANO per aver successo.

STRUTTURA DEL PROGRAMMA

Il programma usa le funzioni PLOT e DRAW per disegnare casualmente il paesaggio ad ogni gioco.

Vengono usati speciali caratteri definibili dall'utente per disegnare la capsula spaziale.

Il test che controlla l'avvenuta collisione della capsula si basa sul test degli attributi.

La struttura del programma è la seguente:

INIZIALIZZAZIONE

DEFINISCI I CARATTERI DEFINIBILI DALL'UTENTE

11: METTI A 0 IL FLAG D'ATTERRAGGIO

VAI A CICLO PRINCIPALE

CICLO PRINCIPALE

DEFINISCI LE VARIABILI DI CONTROLLO SCHERMO
DISEGNA IL PAESAGGIO E IL LUOGO D'ATTERRAGGIO
INIZIALIZZA LA POSIZIONE DELLA CAPSULA

L1: MEMORIZZA LA POSIZIONE VECCHIA

CALCOLA LA NUOVA POSIZIONE

CANCELLA LA VECCHIA FORMA, DISEGNA LA NUOVA

SE C'È IL CRASH

disegna l'esplosione (GOSUB 2000)

genera il suono

vai a E1:

SE NON È ALLUNATO

vai a L1:

LAMPEGGIA IL BORDO, DISEGNA LA BANDIERA, GENERA IL SUONO

E1: PAUSA

VAI A I1:

NOTE SPECIALI

Il tempo che intercorre tra la cancellazione e il disegno della capsula deve essere breve perché questa non lampeggi.

Il colore dello sfondo del display è bianco (7) e l'inchiostro del display è nero (0).

Il controllo per determinare l'eventuale collisione della capsula è basato sul colore degli attributi; per il paesaggio, gli attributi sono INK RED (2), PAPER WHITE (7); per il punto d'atterraggio INK BLUE (1), PAPER CYAN (5).

Dunque per un allunaggio perfetto delle due gambe della capsula, il test corretto sarà (41+41) sugli attributi della nuova posizione per le due gambe inferiori.

Così, se volete cambiare il colore degli attributi del paesaggio o del luogo d'atterraggio o del terreno, bisognerà che sistemiate i valori nel testo nelle linee 8100 & 8150.



ALLUNAGGIO

```
100 REM LUNAR
110 REM Lunar Lander
120 RESTORE : FOR I=1 TO 16: REM inizializza la
    GRAFICA
130 LET S=USR (CHR$ (144+I-1))
140 FOR J=1 TO 8
150 READ g: POKE (S+J-1),g
160 NEXT J
170 NEXT I
180 DATA 0,0,1,1,7,7,31,25,0,0,128,128,224,224,24
    8,152,53,107,127,243,96,96,192,192,172,214,25
    4,207,6,6,3,3
190 DATA 1,1,1,1,1,1,1,1,224,176,140,134,129,135,
    140,176,1,1,7,1,7,7,31,25,224,128,224,128,224
    ,224,248,152
200 DATA 0,0,0,4,5,15,6,15,0,0,0,0,96,192,224,248
    ,27,7,3,6,0,0,0,0,120,168,112,32,0,0,0,0
210 DATA 160,67,38,60,24,48,96,97,4,154,244,124,6
    ,4,4,6,194,96,32,36,104,120,79,193,130,14,12,
    36,124,78,195,2
220 LET ld=0
230 GO TO 8000
1000 REM
1010 REM disegna la base e il  paesaggio
1020 LET bx=INT (RND*21): IF bx<11 THEN GO TO 1020
1030 LET lt=bx*8: LET rt=(bx+4)*8
1040 PRINT AT 21,bx;"{SG8}": PRINT AT 21,bx+1;
    PAPER 5; INK 1;"{SG6}{SG6}": PRINT AT 21
    ,bx+3;"{SG8}"
1050 LET di=-1: GO SUB 1500: REM disegna il paesag
    gio a sinistra
1060 LET di=1: GO SUB 1500: REM disegna il paesagg
    io a destra
1070 RETURN
1500 REM
1510 REM disegna il paesaggio
1520 REM input di
1530 LET ly=8+(di=1)*16
1540 LET lx=((rt+24)*(di=1))+(lt*(di=-1))
1550 LET up=255-(di=-1)*255
1560 IF di=1 THEN PLOT rt,8: DRAW INK 2;24,16
1570 PLOT lx,ly
1580 LET a=RND: LET ry=((a<=.6)-(a>.6))*(INT (RND*48)+1)
1590 LET rx=di*(INT (RND*16)+1)
1600 IF di*(lx+rx)>up*di THEN LET rx=up-lx
1610 IF ly+ry>144 OR ly+ry<0 THEN LET ry=0-ry
1620 LET lx=lx+rx: LET ly=ly+ry
```

```

1630 DRAW INK 2;rx,ry
1640 IF lx<>up THEN GO TO 1550
1650 RETURN
2000 REM
2010 REM Routine esplosione
2020 PRINT AT H,X;"^."
2030 FOR J=1 TO 5
2040 PRINT AT H+1,X;"AB": PRINT AT H+2,X;"CD"
2050 BEEP .05,-(RND*48)
2060 FOR I=1 TO 10: NEXT I
2070 PRINT AT H+1,X;"IJ": PRINT AT H+2,X;"KL"
2080 BEEP .05,-(RND*48)
2090 FOR I=1 TO 10: NEXT I
2100 PRINT AT H+1,X;"MN": PRINT AT H+2,X;"OP"
2110 BEEP .05,-(RND*48)
2120 FOR I=1 TO 10: NEXT I
2130 NEXT J
2140 FOR i=h+1 TO 20: PRINT AT i-1,x;"^.";AT i,x;"
DC";AT i+1,x;"AB": NEXT i
2150 RETURN
8000 REM
8010 REM Ciclo di controllo principale
8020 BORDER 3: INK 0: PAPER 7: OVER 0: FLASH 0:CLS
8030 GO SUB 1000: REM disegna ilpaesaggio
8040 LET X=0: LET H=X
8050 LET OH=H: LET OX=X
8060 LET X=X+0.5*(X<30)-(INKEY$="5")
8070 LET H=H+0.5-(INKEY$="7")*(H>0)
8080 PRINT AT OH,OX;"^." : PRINT AT OH+1,OX;"^."
8090 PRINT AT H,X;"AB": PRINT AT H+1,X;"CD"
8100 LET cr=ATTR (H+2,X)+ATTR (H+2,X+1)
8110 IF cr<58 OR cr=82 OR cr=112 THEN GO TO 8140
8120 GO SUB 2000: REM esplosione
8130 BEEP .5,17: BEEP .5,15: BEEP .25,13: BEEP .25
,12: BEEP .5,10: BEEP .25,13: BEEP .25,12:
GO TO 8210
8140 IF H>=20 THEN GO TO 8120
8150 IF cr=82 THEN LET ld=1: GO TO 8170
8160 IF ld=0 THEN GO TO 8050
8170 FOR I=1 TO 6: PAUSE 25: BORDER I: NEXT I: BORDER 3
8180 PRINT AT 18,bx+1;"EF": PRINT AT 19,bx+1;"GH"
8190 BEEP .125,12: BEEP .25,19: PAUSE 10: BEEP .125,12:
5,12: BEEP .25,19: PAUSE 10: BEEP .125,12:
BEEP .25,19: PAUSE 10: BEEP .125,12: BEEP .2
5,19: PAUSE 10
8200 BEEP .25,12: BEEP .125,14: BEEP .125,15:
BEEP .125,22: BEEP .125,22: BEEP .125,15:
BEEP .125,14: BEEP .125,19: BEEP .125,14:
BEEP .25,12
8210 FOR I=1 TO 100: NEXT I: CLS : GO TO 220

```

Attacco spaziale

Copyright (c) by Clifford Ramshaw

Siete l'unico difensore della stazione spaziale. Improvvisamente li vedete di fronte a voi — un convoglio di nemici alieno. Essi devono sapere del vostro potere perché stanno al sicuro fuori dalla portata del vostro raggio trattore.

Cosa succede? — lentamente uno di loro si separa dal resto della flotta e attacca la vostra posizione. È possibile che stia sbattendo le sue ali oppure è un'illusione dello spazio?

Non preoccupatevi di questo — stanno cadendo missili. Evitatele, quelle armi sono veloci; rapide; Preso. Attento un altro alieno si sta staccando dal convoglio...

CONSIDERAZIONI SUL PROGRAMMA:

Questo programma è un arcade game per lo Spectrum basato su giochi simili che si trovano su altre macchine.

Poiché vanno considerati principalmente la velocità e gli effetti visivi, il suono viene usato solamente quando è necessario dato che, altrimenti, rallenterebbe troppo le cose.

Per lo stesso motivo deve essere fatto il minimo lavoro ad ogni ciclo, per mantenere la velocità al massimo. Devono essere aggiornate solo 3 variabili in ogni ciclo:

coordinate dell'alieno che attacca
posizione della colonna del giocatore

Il ciclo principale è suddiviso in due cicli, uno per disegnare le ali all'alieno e uno per cancellarle. Questo consente al giocatore e all'alieno di muoversi più velocemente poiché non è necessario prendere decisioni all'interno del ciclo principale come quelle circa la forma da disegnare.

I codici di controllo del colore sono usati all'interno delle stringhe PRINT piuttosto che specificando separatamente INK 4, ecc., per velocizzare ancor più le cose.

STRUTTURA DEL PROGRAMMA:

| <u>LINEE</u> | <u>DESCRIZIONE</u> |
|--------------|--|
| 100-340 | Definisce le forme, disegna il convoglio |
| 500-630 | Prima parte del ciclo aggiorna tutte le posizioni |
| 650-790 | Seconda parte del ciclo |
| 1000-1440 | Scontro vincente per il giocatore o per l'alieno |
| 1500-1540 | Distruzione del convoglio nemico |

ATTACCO SPAZIALE

```
100 DATA 14,31,28,24,16,16,16,16,112,248,56,24,8,
      8,8,8: REM forma delle ali chiuse -AB
110 DATA 3,15,63,127,127,127,255,227,192,240,252,
      254,254,254,255,199: REM forma della base sup
      eriore del corpo - CD
120 DATA 227,227,99,99,127,63,7,0,199,199,198,198
      ,254,252,224,0: REM forma della base inferior
      e del corpo - EF
130 DATA 240,112,56,248,124,62,255,30,15,14,28,31
      ,62,124,255,120,7,0,0,0,0,0,0,0,224,0,0,0,0,0
      ,0,0: REM forma delle ali alzate -GHIJ
140 DATA 0,0,0,0,0,0,0,0,7,0,0,0,0,0,0,0,224,30,255
      ,62,124,248,56,112,240,120,255,124,62,31,28,1
      4,15: REM forma delle ali abbassate - KLMN
150 DATA 16,16,16,16,24,28,31,14,8,8,8,8,24,56,24
      8,112: REM forma delle ali cadenti - OP
160 DATA 0,0,12,15,31,63,127,255,24,60,60,255,255
      ,255,231,129,0,0,48,240,248,252,254,255:
      REM forma della base laser -QRS
170 DATA 66,66,38,27,15,7,3,1,66,66,100,216,240,2
      24,192,128: REM forma delle pallottole nemich
      e - TU
180 FOR i=USR "A" TO USR "A"+167
190 READ x: POKE i,x
200 NEXT i
210 BORDER 7: PAPER 7: INK 0: OVER 0: CLS
220 LET v=21
230 LET s=1
240 LET g=5
250 LET x=3+4*s
260 LET y=0
270 PRINT TAB x;
280 FOR i=1 TO 5-s
290 PRINT "CD.";
300 NEXT i
310 PRINT : PRINT TAB x;
320 FOR i=1 TO 5-s
330 PRINT "AEFB.";
185 NEXT i
500 LET g=g+(g<27)*(INKEY$="c")-(g>1)*(INKEY$="z"
)
510 PRINT AT v,g;"QRS.";AT y,x;"....";AT y+1,x
      ;"...."
520 LET x=x+(y>3)-(y<4)+2*SGN (g-x)*(RND>.5):
      IF x>28 THEN LET x=0
530 LET y=y+1
540 PRINT AT y,x;"GCDH";AT y+1,x;"IEFJ"
```

```

550 IF y=v-2 THEN GO TO 1400
560 IF RND>.15 THEN GO TO 600
570 FOR i=y+2 TO v-1: BEEP .02,v-i: PRINT AT i,x+
1;"^";AT i+1,x+1;"TU": NEXT i
580 PRINT AT i,x+1;"^"
590 IF g=x OR g=x-1 THEN GO TO 1300
600 IF INKEY$<>"m" THEN GO TO 650
610 LET m=150: IF g=x OR g=x-1 THEN LET m=157-8*
y
620 PLOT-19+8*g,8: DRAW 0,m: BEEP .1,0: IF m<150
THEN GO TO 1000
630 DRAW INVERSE 1;0,1: DRAW INVERSE 1;0,-m-1
640 REM secondo ciclo
650 LET g=g+(g<27)*(INKEY$="c")-(g>1)*(INKEY$="z"
)
660 PRINT AT v,g;"^QRS";AT y,x;"^";AT y+1,x
;"^"
670 LET x=x+(y>3)-(y<4)+2*SGN (g-x)*(RND>.5):
IF x>28 THEN LET x=0
680 LET y=y+1
690 PRINT AT y,x;"KCDL";AT y+1,x;"MEFN"

700 IF y=v-2 THEN GO TO 1400
710 IF RND>.15 THEN GO TO 750
720 FOR i=y+2 TO v-1: BEEP .01,v-i: PRINT AT i,x+
1;"^";AT i+1,x+1;"TU": NEXT i
730 PRINT AT i,x+1;"^"
740 IF g=x OR g=x-1 THEN GO TO 1300
750 IF INKEY$<>"m" THEN GO TO 500
760 LET m=150: IF g=x OR g=x-1 THEN LET m=157-8*
y
770 PLOT 19+8*g,8: DRAW 0,m: BEEP .1,0: IF m<150
THEN GO TO 1000
780 DRAW INVERSE 1;0,1: DRAW INVERSE 1;0,-m-1
790 GO TO 500
1000 FOR y=y+1 TO v-2
1005 BEEP .02,y/2
1010 PRINT AT y-1,x;"^";AT y,x;"O CDP";
AT y+1,x;"^ EF"
1015 NEXT y
1020 PRINT AT y-1,x;"^"
1200 LET s=s+1
1210 PRINT AT y,x;"^";AT y+1,x;"^";AT 0,0;;
IF s<5 THEN GO TO 250
1220 IF s=5 THEN GO TO 1500
1300 PRINT AT v,g+1;"QRS"; OVER 1;AT v,x+1;"TU":
FOR i=1 TO 11: BEEP .1,-10: PRINT OVER 1;
AT v,g;"^QRS": BEEP .02,4: NEXT i
1310 PAUSE 50
1320 CLS

```

```

1330 RUN 220
1400 PRINT AT y,x;"^";AT y+1,x;"^";AT 0,0;
1410 IF ABS (x-g)>1 THEN GO TO 250
1420 PRINT AT v,g+1;"QRS"; OVER 1;AT y+1,x;"G
CDH";AT y+2,x;"IEFJ": FOR i=1 TO 11:
    BEEP .1,-10: PRINT OVER 1;AT v,g;"QRS":
    BEEP .02,4: NEXT i
1430 PAUSE 50
1440 RUN 100
1500 PRINT AT 5,0;"Congratulazioni - hai distrutto
    l'intera flotta nemica"
1510 BEEP .3,1: BEEP .05,0: BEEP .05,-1: BEEP .5,9
    : BEEP .05,0: BEEP .05,-1: BEEP .6,1: BEEP .5
    ,10
1520 PRINT FLASH 1; PAPER 2; INK 7;AT 10,0;"ATTEN
    TO - STANNO TORNANDO"
1530 FOR i=1 TO 3: BEEP 1,20: BEEP .1,0: NEXT i
1540 RUN 100

```

Spectrum invaders

Copyright (c) by Beam Software

È difficile crederci, ma il più famoso arcade game di tutti i tempi, che introdusse il concetto di arcade games, ed è ancora uno dei più popolari in tutto il mondo, è disponibile sul vostro Spectrum!

Caratteristiche di questa versione dello Spectrum Invaders

- 24 invasori che si muovono da una parte all'altra;
- invasori animati – guardateli muoversi;
- scudi dietro i quali nascondersi
- pericolose bombe nemiche
- potenza di fuoco laser
- aggiornamento continuo del punteggio
- effetti sonori
- la difficoltà aumenta con ogni ondata di invasori

Per giocare, usare i tasti "X" e "C" per muovere il laser verso sinistra e destra, e il tasto "M" per sparare i missili.

La responsabilità di difendere il pianeta è tua, terrestre!

CONSIDERAZIONI SUL PROGRAMMA:

Uno dei problemi nel tentare di scrivere un programma in BASIC per un arcade game è l'alto numero di variabili richieste insieme all'esigenza di un'azione veloce.

Non sarebbe possibile scrivere un programma in BASIC di questo tipo, tenendo conto di 27 paia di variabili (coordinate x, y di 24 invasori, base del laser, bombe nemiche e distruzioni da laser) e avere ancora un ragionevole arcade game. Sarebbe indubbiamente altrettanto difficile averli in movimento e tener conto per esempio degli invasori colpiti.

Questo programma fa uso di due caratteristiche dello Spectrum, che, insieme, rendono possibile questo avvincente programma:

- Stampare una stringa lunga non è molto più lento che stampare una stringa più corta. È certamente molto più veloce che non stampare un certo numero di stringhe.

- È possibile usare l'unica possibilità di slicing sullo Spectrum per assegnare parte di una stringa, per esempio, possiamo dire:
LET a\$ (2 TO 6) = "SALVE"

Il cuore del programma sta in due stringe a\$ e b\$, e nella loro manipolazione. La stringa a\$ è usata per contenere tutti gli invasori, stamparli ecc. Sotto tutti i punti di vista, b\$ è identica a a\$, perché ha sempre la stessa lunghezza, ha altrettanti invasori, ecc., ma b\$ contiene gli stessi invasori in una posizione leggermente diversa. Alternando rapidamente a\$ e b\$, gli invasori sembrano muovere le braccia e le gambe.

Stampando a\$ (e b\$) in posizioni varie, possiamo stampare in un colpo solo tutti gli invasori e tener conto automaticamente del problema della cancellazione (la nuova scritta verrà stampata sopra la vecchia).

Nel caso che l'invasore sia colpito da un laser, viene rimpiazzato da uno spazio all'interno della stringa usando le tecniche di slicing descritte sopra. Regolarmente, inoltre, la fine della stringa viene controllata per vedere se contiene tutti spazi (cioè se tutti gli invasori nella riga in fondo sono stati uccisi). Se è così, lo slicing su stringhe viene usata ancora per accorciare la stringa alla sua nuova lunghezza massima.

Questo non solo evita ogni possibile problema di scrittura oltre il fondo dello schermo, ma dà un controllo immediato anche nel caso che la base sia stata invasa. Una piccola conseguenza di ciò, è che la stringa rimpicciolisce, impiega meno tempo per essere stampata e il gioco accelera un po' verso la fine!

STRUTTURA DEL PROGRAMMA:

| <u>LINEE</u> | <u>DESCRIZIONE</u> |
|--------------|---|
| 100-160 | Definisce i caratteri grafici |
| 200-625 | Inizializzazione Il compito principale è creare a\$ e b\$ usando una subroutine a 9000-9030 |
| 1000-1200 | Prima parte del ciclo principale viene stampata a\$, viene aggiornata l'esplosione, aggiornata la posizione della base laser e se necessario viene aggiornata la posizione della bomba |
| 2000-2999 | Seconda parte del ciclo principale lo stesso come nella prima parte ma con b\$, ancora creazione di nuove bombe nemiche |
| 3000-3530 | Cambio di direzione per gli invasori controllata a\$ per possibile slicing |
| 5000-5220 | Effetto dell'esplosione laser |
| 6000-6530 | Effetto dell'esplosione della bomba nemica |
| 8500-8550 | Fine del gioco |

SPECTRUM INVADERS

```
100 DATA 0,31,63,63,123,113,225,225,0,248,252,252
    ,222,142,135,135,255,127,111,111,199,192,240,
    240,255,254,246,246,227,3,15,15
110 DATA 7,63,127,65,193,199,255,124,224,252,254,
    131,131,227,254,62,124,127,51,48,112,88,136,2
    48,254,204,142,26,17,31,0,0
120 DATA 31,63,63,115,113,225,225,255,248,252,252
    ,206,142,135,135,255,127,111,103,103,96,48,60
    ,60,254,246,230,230,6,12,60,60
130 DATA 7,63,127,193,193,199,127,124,224,252,254
    ,130,131,227,255,62,127,51,113,88,136,248,0,0
    ,62,254,204,12,14,26,17,31
140 DATA 24,24,24,24,60,126,231,231,7,31,63,60,11
    2,112,248,248,224,248,252,60,14,14,31,31
150 DATA 18,26,14,14,7,3,1,0,36,44,56,56,112,96,1
    92,128
160 FOR i=USR "a" TO USR "a"+167: READ x: POKE i,
    x: NEXT i
200 BORDER 7: PAPER 7: INK 0: CLS
210 PRINT AT 0,0;"punti";AT 0,23;"vite ↑↑↑"
220 LET f=2
230 LET sc=0
250 LET q=f: LET f=f+1: IF f=11 THEN STOP
300 LET c$=""
310 LET s=144
320 GO SUB 9000
330 LET k$=a$+b$+c$
340 GO SUB 9000
350 LET l$=a$+b$+c$
360 GO SUB 9000
370 LET m$=a$+b$+c$
380 GO SUB 9000
390 LET n$=a$+b$+c$
400 LET a$=k$+l$+k$+l$: LET a$=a$( TO 343)
410 LET b$=m$+n$+m$+n$: LET b$=b$( TO 343)
500 LET k$="{G4}{SG4}{G3}{G7}{SG7}▲▲▲": LET l$="
    {2SG6}{G3}{2G6}▲▲▲": LET m$="{G7}{G2}{G1}
    {SG4}▲▲▲": LET n$=""
510 PRINT INK 4;AT 16,2;k$;k$;k$;k$;l$;l$;l$;l$;
    m$;m$;m$;m$
590 LET l=0
600 LET r=0: LET s=10: LET t=1
610 LET u=0: LET v=0: LET x=0: LET y=0
615 LET d=21: LET e=12
620 PRINT AT 20,30;c$
625 LET li=3
1000 FOR p=r TO s STEP t
```

```

1010 BEEP .005,1: PRINT AT q,0;c$( TO p);a$;AT 20,
      1;"Q                                R{G3}S."
1020 IF u<q THEN GO TO 1050
1035 IF SCREEN$(u,v)<>" " THEN GO SUB 5000
1040 IF u>=q THEN PRINT AT u,v;"Q";AT u+1,v;"^"
1050 LET 1=1+(1<28)*(INKEY$="c")-(1>0)*(INKEY$="z"
      )
1060 IF u>q-1 OR INKEY$<>"m" THEN GO TO 1080
1070 LET u=20: LET v=1+1
1100 IF d>20 THEN GO TO 1200
1110 IF SCREEN$(d,e)<>"^" THEN GO SUB 6000
1115 IF SCREEN$(d,e+1)<>"^" THEN GO SUB 6000
1120 IF d<=20 THEN PRINT AT d-1,e;"^."; INK 2;
      AT d,e;"TU"
1200 LET u=u-1
2010 PRINT AT q,0;c$( TO p);b$;AT 20,1;"Q
      R{G3}S."
2020 IF u<q THEN GO TO 2050
2035 IF SCREEN$(u,v)<>"^" THEN GO SUB 5000
2040 IF u>=q THEN PRINT AT u,v;"Q";AT u+1,v;"^"
2050 LET 1=1+(1<28)*(INKEY$="c")-(1>0)*(INKEY$="z"
      )
2060 IF u>q-1 OR INKEY$<>"m" THEN GO TO 2080
2070 LET u=20: LET v=1+1
2100 IF d<=20 THEN GO TO 2130
2110 LET w=INT(6*RND): LET x$a$(LEN a$-4*w-2):
      IF x$<>"C" AND x$<>"G" THEN GO TO 2130
2120 LET d=q+1+INT(LEN a$/32): LET e=p-4*w+20+t
2125 IF e=31 THEN LET e=30
2130 IF d=20 THEN PRINT AT 20,e;"^."
2200 LET u=u-1
2210 LET d=d+1
2999 NEXT p
3000 IF t=-1 THEN GO TO 3500
3010 LET t=-1: LET r=9: LET s=0
3020 GO TO 1000
3500 IF LEN a$>55 THEN GO TO 3510
3505 IF a$(23 TO )=c$ THEN GO TO 250
3510 IF a$(LEN a$-32 TO )=c$ THEN LET a$a$( TO
      LEN a$-96): LET b$b$( TO LEN a$)
3520 IF 32*q+LEN a$>609 THEN GO TO 8500
3525 LET r=0: LET s=10: LET t=1
3530 PRINT AT q,0;c$: LET q=q+1: GO TO 1000
5000 IF ATTR(u,v)=60 THEN GO TO 5100
5005 IF ATTR(u,v)=58 THEN GO TO 6000
5010 LET h=v-p+32*(u-q-1)
5015 BEEP .05,8
5020 IF h>=0 THEN GO SUB 5200
5030 LET h=h+32: IF h<LEN a$ THEN GO SUB 5200
5040 LET h=h+32: IF h<LEN a$ THEN GO SUB 5200

```

```

5050 LET h=4-INT (h/96)
5060 LET sc=sc+100*h: PRINT AT 0,6;sc
5100 REM crea un'esplosione
5110 PRINT INK 6;AT u,v;"BA";AT u+1,v;"SR":
      BEEP .05,10: PRINT AT u,v;"▲▲";AT u+1,v;"▲▲"

5120 LET u=0
5130 RETURN
5200 IF h=0 THEN LET h=1
5205 LET a$(h TO h+2)="▲▲▲"
5210 LET b$(h TO h+2)="▲▲▲"
5220 RETURN
6000 REM guarda dove colpiscono i proiettili alien
      i
6005 IF d=20 THEN GO TO 6018
6010 PRINT AT d-1,e;"▲▲"; INK 2;AT d,e;"AB":
      BEEP .05,-10: PRINT AT d,e;"▲▲"
6012 IF (v=e OR v=e+1) AND d=u THEN PRINT AT u,v;
      "▲";AT u+1,v;"▲": LET u=0
6015 LET d=21: RETURN
6020 PRINT AT d-1,e;"▲▲"; INK 2;AT d,e;"AB":
      BEEP .05,10: PRINT AT d,e;"▲▲"
6030 FOR k=1 TO 10: PRINT OVER 1;AT 21,1;"R{G3}S"
      : BEEP .1,0: NEXT k
6040 LET li=li-1
6050 PRINT AT 0,29;"▲▲▲";AT 0,29;"↑↑↑"( TO li)
6060 IF li=0 THEN GO TO 8510
6500 REM i missili si colpiscono fra loro
6510 IF d<=20 THEN PRINT INK 6;AT d,e;"BA";AT d+
      1,e;"SR": BEEP .05,10: PRINT AT d,e;"▲▲";AT d
      +1,e;"▲▲"
6520 LET u=0: LET d=21
6530 RETURN
8500 PRINT AT q,0;c$;a$
8510 PRINT AT 5,0;"▲▲Gli alieni ti hanno invaso.":
      PRINT "▲▲Tutto e' perduto."
8520 INPUT "Vuoi provare ancora?";n$
8530 IF LEN n$=0 THEN GO TO 8520
8540 IF n$(1)="s" OR n$(1)="S" THEN RUN 200
8550 STOP
9000 LET a$=CHR$(s)+CHR$(s+1)+"▲▲": LET a$=a$+a$
      +a$+a$+a$+a$+"▲▲▲▲▲▲▲▲"
9010 LET b$=CHR$(s+2)+CHR$(s+3)+"▲▲": LET b$=b$+
      b$+b$+b$+b$+b$+"▲▲▲▲▲▲▲▲"
9020 LET s=s+4
9030 RETURN
9998 LET u=19: LET v=1: LET d=16: LET e=0:
      CONTINUE
9999 PRINT PEEK 23730+256*PEEK 23731-PEEK 23653-25
      6*PEEK 23654

```


Meteor storm

Copyright (c) by Beam Software

METEOR STORM vi colloca ai controlli di un veicolo spaziale lanciato attraverso la cintura degli asteroidi. Sul vostro display avete la vista piacevole del sistema solare ripreso dallo spazio.

A distanza, potete vedere Saturno con i suoi anelli, e dietro le migliaia di stelle delle galassie vicine.

Improvvisamente i comandi vi avvertono di uno stormo di meteore!

Proprio davanti agli occhi, vedete una meteora ingrandirsi come se si dirigesse direttamente verso la vostra nave! Per confermare la situazione disperata, il vostro radar di bordo segnala l'approccio dell'asteroide.

METEOR STORM è un gioco tridimensionale molto eccitante.

ESECUZIONE DEL PROGRAMMA:

Dopo aver digitato RUN (ENTER) vedrete apparire la consolle.

Un mirino al centro dello schermo segna la direzione nella quale il vostro raggio laser punta. Per muovere il raggio laser, usate i seguenti controlli:

"I" e "P" per muovervi a sinistra o a destra

"W" e "X" per muovervi su e giù

"O" per sparare

Notate che, come avviene per la maggior parte dei programmi BASIC, può essere premuto solamente un tasto alla volta.

Il controllo del mirino ha la caratteristica del "wrap-around" — in altre parole, andare troppo a destra vi porterà alla sinistra dello schermo e così via.

Potete sparare quando volete, ma mancando l'asteroide sentirete solamente il suono sordo del laser che colpisce il vuoto. Colpendo in qualsiasi punto l'asteroide, comunque, lo si vedrà scagliato indietro, e voi sarete fuori pericolo.

Penetrando più profondamente nello STORMO di METEORE, queste incominciano a venire sempre più veloci verso di voi. Quante ne potrete distruggere con sole tre vite?

CONSIDERAZIONI SUL PROGRAMMA:

Questo programma fa un uso estensivo della funzione OVER dello Spectrum e, dimostra la possibilità di riutilizzazione dei caratteri definibili dall'utente.

All'inizio del programma, i caratteri definibili dall'utente vengono usati per disegnare la forma della consolle della nave spaziale, lo schermo radar della nave e il pianeta Saturno. Questo è l'unico punto nell'intero programma in cui essi vengono disegnati! Infatti, subito dopo, quegli stessi caratteri definibili dall'utente vengono riutilizzati per definire la forma degli asteroidi!

Questo funziona perché l'immagine sullo schermo non dipende da questi caratteri — una volta che qualcosa sia stato scritto nella memoria di schermo, in qualunque modo, esso resterà finché qualcosa d'altro non gli venga scritto sopra.

Diviene ora evidente l'immenso vantaggio dell'istruzione OVER.

Per mantenere da vicino il controllo su cosa è stato scritto, e assicurarsi di cancellarlo solo scrivendogli sopra esattamente la stessa cosa allo stesso punto di nuovo, possiamo ottenere tutte le informazioni sullo schermo nel loro formato originale.

(Il modo in cui lavora OVER è quello di scrivere sopra ad una figura la stessa cosa una seconda volta, cosicché questa verrà cancellata, lasciando intatta qualsiasi cosa si trovasse originariamente sotto — provatelo!)

STRUTTURA DEL PROGRAMMA:

| <u>LINEE</u> | <u>DESCRIZIONE</u> |
|--------------|---|
| 100-200 | Definizione dei caratteri come segue: Angoli rotondi Nave spaziale come si vede sul radar Saturno Forme delle meteore piccole I (ABCD) Forme delle meteore piccole II (EFG) Forme delle meteore III (HIJK) Forme delle meteore IV (LM) Forma dell'esplosione I (NOP) Forma dell'esplosione II (QRS) |
| 200-340 | Sistema lo schermo della consolle |
| 400-590 | Definisce ancora i caratteri e inizializzazione. Messaggio lampeggiante. La funzione che richiede più tempo, è la definizione del vettore b\$, che contiene le forme, stabilite a caso, delle meteore usate nell'esplosione del video nella collisione finale. Variabili utili sono: x,y posizione del mirino ox, oy vecchia posizione del mirino li numero di vite ms tipo di meteore (0-9) di livello di difficoltà mx, my posizione delle meteore v colore dell'inchiostro sotto il mirino |

- 900-1100 Ciclo principale
 Controllo della tastiera, muove il mirino
- 1200-1500 Esplosione del laser
- 3000-3100 Subroutine per aggiornare il radar
- 4000-4880 Subroutines per disegnare le meteore nelle varie parti
- 5000-fine Routine per gestire la collisione con la meteora

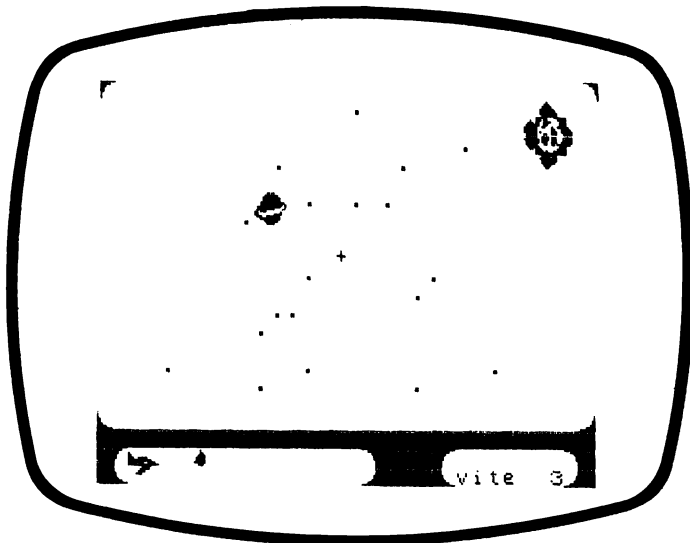
NOTE SPECIALI:

L'esteso uso che fa questo programma delle istruzioni DATA spreca molta memoria, e il programma non girerebbe in uno Spectrum 16K se non si prendessero misure speciali per ridurre l'uso della memoria.

Per esempio, per ogni numero in un'istruzione DATA, non vengono memorizzati soltanto i caratteri che vedete, ma anche una sequenza di 6 bytes che è in un formato tale da consentire alla ROM dello Spectrum di operare agevolmente. In altre parole, una stringa DATA con 32 numeri, occupa 6 bytes per il numero di linea e le altre informazioni, circa 100 bytes per i caratteri che vi compaiono e ulteriori 200 bytes per il modo in cui il computer memorizza queste informazioni.

Il costo di 6 bytes è presente in tutte le istruzioni BASIC, non solo nelle DATA. Noterete perciò che questo programma definisce i due numeri più comunemente usati, 0 e 1, proprio all'inizio, come variabili, e che queste vengono usate in tutto il programma.

Il risparmio di memoria è enorme, ogni numero richiede ora soltanto 2 bytes, includendo la virgola che separa i numeri.



METEOR

```
90 LET z=0: LET u=1
100 DATA 255,248,224,192,192,128,128,128,255,31,7
,3,3,u,u,u,128,128,128,192,192,224,248,255,u,
u,u,3,3,7,31,255
110 DATA z,z,128,192,231,255,255,255,z,z,z,z,192,
240,60,255,u,3,7,14,z,z,z,z,240,224,128,z,z,z
,z,z
120 DATA z,3,7,15,15,31,31,127,z,192,240,248,248,
254,253,249,127,220,192,115,31,7,3,z,195,4,60
,248,248,240,192,z
130 DATA z,z,u,3,7,7,7,2,z,4,6,15,31,31,63,14,z,4
,14,31,63,63,63,31,30,30,28,12,z,z,z,z
140 DATA 31,63,31,31,15,6,z,z,z,z,z,128,192,224
,224,224,224,192,z,z,z,z
150 DATA 3,23,55,63,127,127,127,127,128,224,240,2
40,224,248,248,252,127,63,63,127,63,31,11,z,2
52,252,248,224,240,224,128,z
160 DATA 63,127,255,127,255,255,127,63,252,248,25
2,254,254,248,248,252
170 DATA 195,153,60,102,102,60,153,195,195,99,39,
60,60,180,231,231,128,60,230,255,91,200,221,1
19
180 DATA 38,62,246,254,108,71,227,62,115,55,246,1
80,28,188,246,99,221,118,103,238,184,115,255,
128
200 BORDER 5: INK 5: PAPER 6: CLS
210 RESTORE 100
220 FOR i=USR "a" TO USR "m"-u
230 READ x: POKE i,x: NEXT i
240 DIM s(10)
300 PRINT AT z,z;"A";AT z,31;"B";AT 18,z;"C";AT 1
8,31;"D"
310 PRINT "{32SG8}": PRINT "{SG8}A.....
B{4SG8}A..... B{2SG8}C.....D
{4SG8}C.....D{SG8}"
320 PRINT INK z;AT 20,2;"EF";AT 21,2;"GH";AT 21,
23;"vite.3"
330 FOR i=u TO 20: PRINT INK 2;AT u+16*RND,u+29*
RND;".": NEXT i
340 PRINT INK 3;AT 6,10;"IJ";AT 7,10;"KL"
400 FOR i=USR "a" TO USR "t"-u
410 READ x: POKE i,x: NEXT i
420 PRINT OVER u; FLASH u; INK 2;AT 10,4;"PREPAR
ATI A VOLARE IN UNO";AT 11,8;"STORMO DI METEO
RE"
430 LET b$="": FOR j=u TO 17
435 BEEP .05,z
```



```

440 LET a$="▲"
450 FOR i=u TO 15
460 IF RND>.6 THEN LET a$=a$+CHR$(144+12*RND)
470 LET a$=a$+"▲▲"
475 BEEP .005,10
480 NEXT i
490 LET b$=b$+a$( TO 29)+"▲": NEXT j
500 PRINT OVER u; FLASH z; AT 10,4;"PREPARATI A V
OLARE IN UNO"; AT 11,8;"STORMO DI METEORE"
520 LET ox=9: LET oy=15: LET ov=5
530 LET x=9: LET y=15: LET v=7
540 INK z: PRINT AT x,y;"+"
550 LET li=3: LET di=u: LET nm=z
560 LET ms=z: LET nm=nm+u
580 LET mx=u+RND*17: LET my=u+RND*29
590 LET mi=u
900 LET a$=INKEY$
910 LET ms=ms+di/7+nm/50
920 IF INT ms=9 THEN GO TO 5000
930 GO SUB 3000
940 GO SUB 4000+100*INT ms
950 LET w=ATTR (x,y)-8*INT (ATTR (x,y)/8): IF w
<>0 THEN LET v=w
960 IF a$="" OR a$="0" THEN GO TO 1070
1000 LET y=oy+(a$="p")-(a$="i")
1010 IF y<u THEN LET y=30
1020 IF y>30 THEN LET y=u
1030 LET x=ox+(a$="x")-(a$="w")
1040 IF x<u THEN LET x=17
1050 IF x>17 THEN LET x=u
1060 LET v=ATTR (x,y)-8*INT (ATTR (x,y)/8)
1070 PRINT OVER u; INK ov; AT ox,oy;"+"; INK z;
AT x,y;"+"
1090 LET ox=x: LET oy=y: LET ov=v
1100 IF a$<>"0" THEN GO TO 900
1200 FOR i=u TO 6: BEEP i*i/100,10/i: NEXT i
1250 IF v<>u THEN GO TO 900
1260 LET x1=x-u: IF x1<z THEN LET x1=z
1270 LET y1=y-u: IF y1<z THEN LET y1=z
1275 LET x2=x+u: IF x2>18 THEN LET x2=18
1280 OVER u: FOR i=u TO 10: BEEP .03,3: PRINT AT x
,y;"N": BEEP .01,u: PRINT AT x1,y1;"OPO"; AT x
,y1;"QNO"; AT x2,y1;"RSR": NEXT i: OVER z
1290 DIM s(10): LET mi=2
1300 FOR i=INT ms TO z STEP -u
1310 LET ms=i: GO SUB 4000+100*i: GO SUB 3000
1320 NEXT i
1340 DIM s(10)
1360 LET ov=z: LET v=z
1370 PRINT INK z; OVER z; AT 20,23;nm+li-3; AT 20,26;"shot"

```

```

1380 PRINT OVER u; INK 3;AT 6,10;"_";AT 7,10;"_."
1500 GO TO 560
3000 IF INT ms<4 THEN PRINT INK u;AT 20,14-ms;"_B."
3010 IF INT ms=4 THEN PRINT INK u;AT 20,9;"_B."
3020 IF INT ms>4 AND INT ms<8 THEN PRINT INK u;
AT 20,18-2*INT ms;"_B."
3030 IF INT ms=8 THEN PRINT INK u;AT 20,4;"_B."
3040 RETURN
4000 IF s(1)=u THEN RETURN
4010 PRINT OVER u; INK mi;AT mx,my;"_A"
4020 LET s(1)=u
4030 RETURN
4100 IF s(2)=u THEN RETURN
4110 PRINT OVER u; INK mi;AT mx,my;"_B"
4120 LET s(2)=u
4130 RETURN
4200 IF s(3)=u THEN RETURN
4210 PRINT OVER u; INK mi;AT mx,my;"_BF"
4220 LET s(3)=u
4230 RETURN
4300 IF s(4)=u THEN RETURN
4310 LET mx4=mx+u
4320 PRINT OVER u; INK mi;AT mx,my;"_BF";AT mx4,my
;"_D"
4330 LET s(4)=u
4340 RETURN
4400 IF s(5)=u THEN RETURN
4410 PRINT OVER u; INK mi;AT mx,my;"_C";AT mx4,my;
"_E"
4420 LET s(5)=u
4430 RETURN
4500 IF s(6)=u THEN RETURN
4510 PRINT OVER u; INK mi;AT mx,my;"_CF";AT mx4,my
;"_EG"
4520 LET s(6)=u
4530 RETURN
4600 IF s(7)=u THEN RETURN
4610 PRINT OVER u; INK mi;AT mx,my;"_HI";AT mx4,my
;"_JK"
4620 LET s(7)=u
4630 RETURN
4700 IF s(8)=u THEN RETURN
4710 LET mx2=mx-u
4720 LET my6=my+2: IF my6>31 THEN LET my6=31
4730 LET mx5=mx+2: IF mx5>18 THEN LET mx5=18
4740 LET my3=my-u
4750 PRINT OVER u; INK mi;AT mx2,my;"_CF";AT mx,my
3;"_C{2SG8}";AT mx,my6;"_F";AT mx4,my3;"_E
{2SG8}";AT mx4,my6;"_G";AT mx5,my;"_EG"
4760 LET s(8)=u

```

```

4770 RETURN
4800 IF s(9)=u THEN RETURN
4810 LET mx1=mx-2: IF mx1<z THEN LET mx1=z
4820 LET my2=my-2: IF my<z THEN LET my=z
4830 LET my7=my+3: IF my7>31 THEN LET my7=31
4840 LET mx6=mx+3: IF mx6>18 THEN LET mx6=18
4850 LET mx6=mx+3: IF mx6>18 THEN LET mx6=18
4860 PRINT OVER u; INK mi;AT mx1,my;"CF";AT mx2,m
y3;"H{2SG8}";AT mx2,my6;"I";AT mx,my2;"C";
AT mx,my3;"{3SG8}";AT mx,my6;"{SG8}";AT mx,my
7;"F";AT mx4,my2;"E";AT mx4,my3;"{3SG8}";AT m
x4,my6;"{SG8}";AT mx4,my7;"G";AT mx5,my3;"J
{2SG8}";AT mx5,my6;"K";AT mx6,my;"EG"
4870 LET s(9)=u
4880 RETURN
5000 FOR i=z TO 16
5010 PRINT OVER u; FLASH u;AT i+u,u;b$(30*i+u
TO 30*i+30)
5020 NEXT i
5030 PRINT FLASH u; OVER u;AT 20,1;"
";AT 21,1;" "
5400 DIM s(10)
5500 FOR i=u TO 40: BEEP RND/10,10*RND-5
5510 IF i>31 THEN GO SUB 8000-100*i
5520 NEXT i
5530 LET ov=4: LET v=4
6000 FOR i=z TO 16
6010 PRINT OVER u; INK 2; FLASH z;AT i+1,1;b$(30*
i+1 TO 30*i+30)
6020 NEXT i
6030 PRINT FLASH z; INK 5; OVER 1;AT 20,1;"^
";AT 21,1;"^
"
6035 PRINT OVER u;AT 20,4;"B"
6040 LET li=li-u
6050 PRINT OVER z; INK z;AT 21,29;li
6060 IF li=z THEN STOP
6070 DIM s(10)
6200 GO TO 560

```


Eliminator

Copyright (c) by Beam Software

Siete il difensore degli umanoidi sulla superficie del pianeta. Il vostro compito è ELIMINARE gli invasori.

Dovete impedire a tutti i costi i loro tentativi di rapire gli umanoidi sotto la vostra protezione. La loro sopravvivenza e la sopravvivenza del loro pianeta dipende da voi.

Il pianeta che state custodendo è molto piccolo ed è possibile volargli attorno molto velocemente, ma potete viaggiare solo in una direzione.

I rapitori nemici hanno sganciato un missile autocercante contro di voi per distrarre la vostra attenzione dalla vera missione. Potete eseguire un'azione diversiva, sparare agli invasori e proteggere gli umanoidi?

ESECUZIONE DEL GIOCO

Siete al controllo del veicolo spaziale nel centro dello schermo. La velocità acquisita in avanti non vi permette alcuna flessibilità nei movimenti laterali.

Pertanto, potete soltanto muovervi sù e giù, usando i tasti "W" e "X". Avete a vostra disposizione un laser distruttore, controllato premendo il tasto "P", che disintegrerà i missili autocercanti e gli invasori rapitori, ma ciò porta soltanto un sollievo transitorio. Non appena ne distruggete uno, ne appare un altro. Il vostro veicolo può sopportare qualche danno, ma dopo 3 colpi non è più operativo.

CONSIDERAZIONI SUL PROGRAMMA

Non ci sono istruzioni disponibili nel BASIC per simulare il movimento del veicolo spaziale contro la superficie del pianeta. Anche se ci fossero state istruzioni adatte (del tipo di SCREEN\$; notate che questa non può essere usata dato che non riconosce i caratteri grafici definibili dall'utente!!!), lo scroll orizzontale dello schermo sarebbe stato penosamente lento.

L'unica soluzione adatta, quella usata qui, è di avere una routine di scroll orizzontale scritta in linguaggio macchina.

Troverete questa routine, lunga 36 bytes, nell'istruzione DATA alla linea 3000.

Ciò che fa questa routine è fare eseguire uno scroll a ciascuna linea dello schermo, con il sistema del "wrap around": cioè il carattere che era a 0,1 andrà a 0,0, il carattere che era a 0,0 andrà a 0,31 e così via.

La routine, poi, fa la stessa cosa per gli attributi, così che il colore associato ad ogni forma si muove con essa.

Questa routine in linguaggio macchina è pienamente rilocabile; in altre parole, non è necessario che sappiate alcunché sul linguaggio macchina per usarla, usate

semplicemente un programma simile a quello alle linee 140 e 150. Poiché questa routine è pienamente rilocabile, potete specificare qualsiasi indirizzo nel quale inserirla.

Potreste voler provare ad adattare questa routine per altri programmi. Siate certi che il Gruppo Editoriale Jackson è sempre interessato a esaminare qualsiasi programma emozionante voi sviluppate.

Il resto del programma consiste nell'aggiornamento delle variabili relative all'astronave (SX, SY), agli alieni (AX, AY) e alle bombe (BX, BY). A causa del continuo scroll, questo fa perdere tempo, ma è essenziale.

STRUTTURA DEL PROGRAMMA

| <u>LINEE</u> | <u>DESCRIZIONE</u> |
|--------------|---|
| 100-150 | Inizializza le variabili, definisce i caratteri |
| 160-360 | Disegna il paesaggio, introduce gli uomini nelle torri |
| 1000-1020 | Disegna la nave, gli alieni e le bombe |
| 5000-5360 | Ciclo principale Controlla il movimento della nave Fa cercare agli alieni gli umanoidi Fa dirigere le bombe sulle navi Spara il laser se il tasto è premuto Controlla eventuali collisioni |
| 7000-7330 | Ciclo rapitore Controlla se rimane qualche umano, ecc. |
| 8500-8760 | Routine del danno alla nave Include la fine del programma |

ELIMINATOR

```
100 RESTORE 2000
110 FOR i=USR "a" TO USR "k"-1
120 READ x: POKE i,x
130 NEXT i
140 RESTORE 3000: CLEAR 32500
150 FOR i=32500 TO 32535: READ x: POKE i,x:
    NEXT i
160 BORDER 1: PAPER 1: INK 0: CLS
170 PLOT 0,8
180 LET y=8
190 LET p=-1
200 FOR j=1 TO 3
210 LET a=3+INT (7*RND)
220 GO SUB 8100
230 DRAW 4,15-y: DRAW 8,0
240 LET y=15
```

```

250 LET p=p+a+1
260 FOR k=1 TO INT (4*RND)+1: PRINT INK 5;
    OVER 1;AT 20-k,p;"C": NEXT k: PRINT INK 3;
    AT 20-k,p;"B";AT 19-k,p;"A"
270 NEXT j
280 LET a=30-p
290 GO SUB 8100
300 DRAW 11,8-y
310 LET sx=3: LET sy=10
320 LET ax=INT (15*RND): LET ay=26
330 LET bx=INT (RND*15): LET by=30: IF bx=ax
    THEN GO TO 330
340 LET t=0
350 LET e=0: LET h=3
360 LET dm=0: LET sc=0
1000 PRINT INK 1;AT sx,sy;"FGHI"
1010 PRINT INK 4;AT ax,ay;"DE"
1020 PRINT INK 2;AT bx,by;"J"
2000 DATA 1,57,189,187,146,254,124,124,56,56,56,1
    04,76,68,68,198: REM forma degli umanoidi
2010 DATA 126,36,60,24,60,102,195,255: REM shape o
    f platform
2020 DATA 63,99,193,193,127,28,7,3,252,198,131,131
    ,254,56,224,192: REM forma degli alieni
2030 DATA 255,63,15,7,3,7,255,0,192,240,252,255,25
    5,255,240,3,0,0,31,249,224,255,252,240,0,0,0,
    192,248,255,0,0: REM forma delle astronavi
2040 DATA 28,28,244,199,227,47,56,56
3000 DATA 33,1,64,17,0,64,26,1,31,0,237,176,18,35,
    19,62,88,188,32,242,6,22,197,26,1,31,0,237,17
    6,18,35,19,193,16,243,201
5000 LET v=USR 32500
5010 LET x$=INKEY$
5020 LET nx=sx+(x$="x")*(sx<18)-(x$="w")*(sx>0)
5030 IF nx=sx THEN GO TO 5060
5040 LET v$=SCREEN$(nx,sy+1): LET w$=SCREEN$(nx,
    sy+2): LET x$=SCREEN$(nx,sy+3): LET v$=v$+w$
    +x$
5050 IF v$<>"..." THEN GO TO 8500
5060 IF SCREEN$(nx,sy+4)<>"." THEN GO TO 8500
5070 PRINT OVER 1;AT sx,sy-1;"FGHI";AT nx,sy;"
    FGHI"
5080 LET sx=nx
5090 LET by=by-1: IF by<0 THEN LET by=31
5100 IF RND>.7 OR by=31 THEN GO TO 5140
5110 LET nx=bx+(sx-1>bx)-(sx-1<bx)+(by=sy+2)
5120 PRINT OVER 1;AT bx,by;"J";AT nx,by;"J"

5130 LET bx=nx
5140 IF bx=sx AND ABS (by-sy-1)<=1 THEN GO TO 8500

```

```

5200 IF x$<>"p" THEN GO TO 5290
5210 BEEP .1,4
5220 PLOT INK 3; OVER 1;8*sy+43,170-8*sx
5230 DRAW INK 3; OVER 1;212-8*sy,0
5240 BEEP .1,8
5250 PLOT INK 0; OVER 1;8*sy+43,170-8*sx
5260 DRAW INK 0; OVER 1;212-8*sy,0
5270 IF ax=sx AND ay>sy+4 AND ay<30 THEN FOR k=1
    TO 11: BEEP .1,5: PRINT OVER 1;AT ax,ay-1;"DE":
    NEXT k: LET ax=INT (15*RND): LET ay=29:
    PRINT AT ax,ay-1;"DE": LET sc=sc+100
5280 IF bx=sx AND by>sy+4 AND by<30 THEN FOR k=1
    TO 5: BEEP .1,10: PRINT OVER 1;AT bx,by;"J":
    NEXT k: LET bx=1+INT (RND*14): LET by=
    30: PRINT AT bx,by;"J": LET sc=sc+20
5290 LET ay=ay-1: IF ay<0 THEN LET ay=31
5300 IF RND>.7 OR ay>=29 THEN GO TO 5000
5310 LET nx=ax+(ax<18)-t: IF ax=18 THEN GO TO 533
    0
5320 LET v$=SCREEN$ (nx,ay+1): LET w$=SCREEN$ (nx,
    ay+2): IF v$+w$<>"AA" THEN GO TO 7000
5330 LET ny=ay+(ax=18)
5340 IF ny<>ay AND SCREEN$ (nx,ny+2)<>"A"
    THEN GO TO 7000
5350 PRINT OVER 1;AT ax,ay;"DE";AT nx,ny;"DE"

5360 LET ay=ny: LET ax=nx
6000 GO TO 5000
7000 IF ABS (bx-ax)<=1 AND ABS (by-ay)<=1 THEN
    FOR k=1 TO 11: BEEP .1,5: PRINT OVER 1;AT a
    x,ay;"DE": NEXT k: LET ax=INT (15*RND):
    LET ay=29: PRINT AT ax,ay;"DE"
7010 IF ABS (sx-ax)<=2 AND ABS (by-sy)<=2 THEN
    GO TO 8500
7020 PRINT OVER 1;AT ax,ay;"DE";AT ax-1,ay;"D
E"
7030 LET ax=ax-1: LET ay=ay-1: LET by=by-1
7040 IF ay<0 THEN LET ay=31
7050 IF by<0 THEN LET by=31
7060 LET v=USR 32500
7070 PRINT OVER 1;AT sx,sy-1;"FGHI";AT sx,sy;"
FGHI"
7080 LET f=0
7090 FOR x=ax-3 TO ax+3
7095 IF x>20 THEN GO TO 7125
7100 FOR y=ay+1 TO ay+3
7105 IF y<0 OR y>31 THEN GO TO 7120
7110 IF ATTR (x,y)=11 THEN PRINT AT x,y; INK 1;"
    ": LET f=f+1
7120 NEXT y

```



```

7125 NEXT x
7130 IF f=2 THEN GO TO 7200
7140 PRINT AT ax,ay+3; INK 1;" ";AT ax+1,ay+2;"▲▲
";AT ax+2,ay+2;"▲▲"
7145 LET ay=ay+1: PRINT OVER 1;AT ax,ay-1;"DE";
AT ax,ay;"DE"
7150 LET e=e+1: IF e=3 THEN GO TO 8550
7160 GO TO 5000
7200 PRINT OVER 1;AT ax,ay;"DE";AT ax-1,ay;"D
E"
7210 LET ax=ax-1: REM LET ay=ay-1: IF ay<0 THEN
LET ay=31
7220 LET by=by-1: IF by<0 THEN LET by=31
7230 LET v=USR 32500
7240 PRINT INK 4;AT ax+1,ay;"A";AT ax+2,ay;"B"
7250 FOR k=ax-1 TO 0 STEP -1
7260 PRINT OVER 1; INK 4;AT k+1,ay;"DE";AT k+2,ay
;"A";AT k+3,ay;"B";AT k,ay;"DE";AT k+1,ay;"
A";AT k+2,ay;"B"
7270 BEEP .2,15-k/2
7280 NEXT k
7290 LET e=0: LET h=h-1: IF h=0 THEN GO TO 8550
7300 PRINT OVER 1;AT sx,sy-1;"FGHI";AT sx,sy;"
FGHI"
7310 PRINT INK 1;AT 1,ay;"▲";AT 2,ay;"▲"
7320 LET ax=0: LET ay=ay-1: IF ay<0 THEN LET ay=3
1
7330 GO TO 5000
8100 FOR i=0 TO 2*a-2
8110 LET ly=INT (36*RND)-18
8120 IF y+ly<0 THEN LET ly=-y
8130 IF y+ly>22 THEN LET ly=22-y
8140 DRAW 4,ly
8150 LET y=y+ly
8160 NEXT i
8170 RETURN
8500 PRINT INK 5; FLASH 1;AT 21,0;"Danni alla nav
e"
8510 LET dm=dm+1: IF dm<3 THEN GO TO 8700
8520 PRINT AT 21,15; FLASH 1; INK 7;" IRREPARABILI
": FOR s=22944 TO 23104
8530 IF PEEK s=11 THEN BEEP .5,10: POKE s,9:
POKE s+32,9
8540 NEXT s
8550 FOR k=1 TO 15
8560 PRINT OVER 1;AT sx,sy-1;" FGHI": BEEP k/50
,12-2*k: NEXT k
8570 PRINT AT 1,0; INK 7;"I sistemi di supporto um
ani non sono piu' operativi. Avete fallito ne
l difendere il pianeta."

```

```

8580 PRINT INK 2;AT 10,0;"Punteggio= ";sc
8600 INPUT "Vuoi tentare ancora? ";a$
8610 LET a$=a$+" ": IF a$(1)="s" THEN RUN 100
8650 STOP
8700 FOR k=1 TO 15
8710 PRINT OVER 1;AT sx,sy-1;"          FGHI": BEEP .1,1
      5-2*k: NEXT k
8720 PRINT INK 1; FLASH 0;AT 21,0;"
      "
8730 PRINT OVER 1;AT ax,ay-1;".DE";AT bx,by-1;".J
      "
8740 LET ax=INT (15*RND): LET ay=26
8750 LET bx=INT (15*RND): LET by=30: IF bx=ax
      THEN GO TO 8750
8760 GO TO 1000
9999 BORDER 7: PAPER 7: INK 0: CLS

```

Freeway Frog

Copyright (c) Beam Software

Siete una povera, piccola rana che cerca disperatamente di andare a casa attraversando l'autostrada. Nei vostri viaggi dovete affrontare autocarri, motocicli, automobili e, soprattutto le auto della polizia lanciate ad alta velocità che possono apparire all'improvviso.

Per controllare la rana usate i seguenti tasti:

"Q" per muovere in su

"A" per muovere in giù

"P" per muovere a destra

"O" per muovere a sinistra

Guadagnate punti andando in su e in giù, così per procurarvi un punteggio alto potete attraversare l'autostrada quante volte volete.

Quando una rana raggiunge il margine superiore dello schermo è a casa e allora dovete muovere la prossima per tentare di farle raggiungere casa.

COME LAVORA IL PROGRAMMA

Questo programma fa largo uso del set di caratteri definibili dall'utente dello Spectrum. Se non sapete di cosa si tratta, vi potrebbe aiutare leggere il capitolo del manuale relativo ad essi. Solamente lo SPECTRUM vi permette di avere contemporaneamente 21 caratteri definibili dall'utente. Questo programma, comunque, ne richiede 58.

Per permettere al computer di accedere a tutti questi caratteri è necessario che l'area UDG venga spostata avanti e indietro tra due o più serie di caratteri. Per far questo, per prima cosa dobbiamo fare posto ai caratteri richiesti. Per un sistema a 16K, con solo 21 caratteri definibili dall'utente, l'area UDG è situata alle locazioni di memoria da 32.600 a 32.767. Questo significa 168 bytes di memoria o, dati 8 bytes per ciascun carattere, 21 caratteri. Siccome abbiamo bisogno di 58 caratteri per il nostro gioco, dobbiamo avere 464 bytes di memoria per l'area UDG, il che corrisponde alle locazioni da 32.304 a 32.767; così, la prima istruzione nel programma deve essere un

```
CLEAR 32.303
```

(NOTA: se viene dato CLEAR dopo che sia stata definita qualsiasi variabile in una LET o una DIM, esse andranno perdute)

In questo programma abbiamo 3 gruppi di caratteri definiti in modo speciale:
I primi due hanno 21 caratteri ciascuno e il terzo solamente 16.

Così per accedere ad ogni gruppo tutto ciò che dobbiamo fare è puntare la variabile di sistema UDG (locazione 23675 e 23676) al blocco corretto. Questi sono:

32.600 per il primo set (il set normale)

32.432 per il secondo set (32.600 - 168)

32.304 per il terzo set (32.432 - 128)

NOTA: il terzo set si sovrappone al secondo perché esso ha solamente 16 caratteri.

Per definire il valore corretto per UDG è necessario qualche calcolo. Il numero da inserire con POKE in UDG è ottenuto da:

POKE 23675, 'valore'-256* INT('valore'/256)

e

POKE 23676, INT('valore'/256)

NOTA: non è necessario mettere effettivamente queste linee nel programma giacché possiamo calcolare noi i valori corretti e usarli.

Esempio

| <u>PER OTTENERE</u> | <u>POKE 23675</u> | <u>POKE 23676</u> |
|---------------------|-------------------|-------------------|
| 32.600 | 88 | 127 |
| 32.432 | 176 | 126 |
| 32.304 | 48 | 126 |

I valori per i caratteri grafici definibili dall'utente devono essere in qualche modo inseriti nel computer e siccome abbiamo 58 caratteri a 464 bytes, dobbiamo inserire 464 valori. Se questi fossero inseriti in istruzioni DATA occuperebbero molto spazio (così tanto, infatti, che non ci sarebbe abbastanza posto per il gioco); e sarebbe sciocco inserirli con l'istruzione INPUT ogni volta che il gioco viene caricato. Così, per superare questo problema, il programma è stato diviso in due parti. La prima costruisce i caratteri grafici definibili dall'utente e, quando questi sono corretti, li salva come blocco di bytes. Quando si dà il RUN al gioco questo blocco di bytes viene caricato dal nastro e il gioco può iniziare.

(Notate che avete bisogno di caricare il blocco di bytes solamente quando il gioco è eseguito la prima volta dopo essere stato caricato dal nastro. Una volta che i caratteri si trovano nel computer, vi rimangono fino a che viene tolta la corrente o un altro programma occupa le stesse locazioni).

DEFINIZIONE DEI CARATTERI GRAFICI DI FREEWAY FROG

Il seguente programma costruirà i caratteri grafici per il gioco. È importante fare molta attenzione quando battete le istruzioni DATA, poiché è molto difficile trovare in seguito eventuali errori. Quando viene dato il RUN, il programma metterà tutti i dati nell'area dei caratteri grafici definibili dall'utente, poi scriverà sullo schermo tutte le forme usate.

Vedrete sullo schermo.

In alto 4 rane verdi.

Le rane dovrebbero apparire uguali, salvo che dovrebbero dirigersi in direzioni diverse (la prima in alto, poi a destra, in giù e l'ultima a sinistra).

Sotto queste ci dovrebbe essere una motocicletta bianca, un'automobile rossa e un autocarro, tutti rivolti a sinistra.

Sotto questi, una motocicletta bianca, un'automobile gialla e un autocarro, tutti rivolti a destra.

Se vi sembra tutto corretto, il set di caratteri può essere salvato su nastro. Usare l'istruzione

SAVE "forme rane" CODE 32304,464

(il nome non ha importanza)

Non dimenticate di salvare anche il programma, nel caso vi possa servire ancora.

PROGRAMMA PRINCIPALE PER FREEWAY FROG

Il programma principale è abbastanza lungo, così potrebbe essere una buona idea digitarlo in diversi blocchi, stando molto attenti e controllando ciò che è stato inserito.

In particolare, le sezioni del programma che disegnano le forme degli oggetti contengono molti caratteri grafici e codici di controllo colore. Poiché questi sembreranno molto diversi, una volta che il programma sia stato eseguito (cioè il carattere grafico A non sarà più una 'A' una volta che i caratteri saranno stati caricati), è molto importante inserirli attentamente. Dopo, è quasi impossibile trovare gli errori.

Quando il programma parte, aspetterà di caricare le forme dei caratteri grafici da nastro. Dovreste mettere nel registratore il nastro con i bytes che avete salvato dal programma precedente e avviarlo. Se volete far partire il programma e i bytes sono già caricati, usate RUN 26.

Segue la descrizione del gioco.

STRUTTURA DEL PROGRAMMA

| <u>LINEE</u> | <u>DESCRIZIONE</u> |
|--------------|--|
| 20-25 | Definisce RAM TOP per far posto ai caratteri grafici definibili dall'utente e poi provvede a caricarli dal nastro. |

- 26-75 Inizializza tutte le variabili usate, definisce i colori dello schermo, le posizioni delle auto, degli autocarri e delle motociclette e le variabili usate per cambiare il set di caratteri UDG (il programma dovrebbe essere fatto partire da qui se i caratteri definibili dall'utente sono già stati caricati)
- 80-100 Disegna l'autostrada
- 105 Disegna le rane rimaste sullo schermo
- 106 Scrive il punteggio e il punteggio massimo
- 110-150 Decide se deve passare un'auto della polizia in mezzo all'autostrada. Se sta per apparirne una, allora viene definita la direzione del suo movimento e quindi questa viene visualizzata.
- 500-530 Controllo per vedere se la rana è stata schiacciata sull'autostrada. Se è così, il programma salta alla linea 4000.
- 600-700 Se il giocatore può muovere (può muovere soltanto una volta ogni 2 ricorsoni del ciclo principale) viene controllata la tastiera per vedere se vuole muovere. Se è così, viene modificata la posizione della rana. La sua direzione viene cambiata, se necessario, e il punteggio è incrementato se si è mossa in su o in giù. La vecchia rana viene allora cancellata e la nuova disegnata.
- 750 Muove tutte le motociclette e le ridisegna.
- 800-820 Ogni secondo ciclo muove tutte le auto e le ridisegna.
- 900-950 Ogni quarto ciclo muove tutti gli autocarri e li ridisegna. Poi salta a 106 per un altro ciclo.
- 990-999 Rimuove una rana dallo schermo, viene controllata la posizione della rana per vedere se l'autostrada deve essere ridisegnata nel posto in cui si trovava la rana
- 1000-1070 Contiene quattro subroutines usate per disegnare le rane rivolte nelle quattro diverse direzioni.
- 1080-1117 Queste 6 subroutines disegnano le automobili, gli autocarri e le motociclette
- 2000-2001 Disegna tutte le automobili sullo schermo
- 2010-2012 Aggiorna le posizioni di tutte le automobili (cioè le auto che si muovono a destra hanno la loro posizione incrementata e quelle che si muovono verso sinistra decrementata).
- 2100-2112 Disegna tutte le motociclette e aggiorna le loro posizioni.
- 2200-2201 Disegna tutti gli autocarri.
- 2210-2212 Aggiorna le posizioni di tutti gli autocarri.
- 4000-4010 Indica al giocatore che una rana è stata investita, poi diminuisce il numero di rane rimaste. Se non ne sono rimaste, il programma torna indietro per ricominciare con la prossima rana.
- 4020-4070 Stampa il messaggio di fine gioco, aggiorna il punteggio massimo, se è necessario, e chiede al giocatore se vuole giocare ancora.
- 5000-5020 Viene fatto un salto qui se una rana arriva a casa, viene diminuito il numero di rane rimaste e, se ne sono rimaste, il gioco continua, altrimenti il programma va a 4020.


```

100 PRINT INK 7;AT 15,3;" Caratteri memorizzati
:*****Carica ora il programma"
120 STOP
9000 DATA z,1,35,37,111,79,223,x
9001 DATA z,128,196,164,246,242,251,x
9002 DATA 111,15,31,159,220,216,120,48
9003 DATA 246,240,248,249,59,27,30,12
9004 DATA 56,113,193,252,127,31,31,31
9005 DATA 192,240,156,192,240,248,244,254
9006 DATA 31,31,31,127,252,193,113,56
9007 DATA 254,244,248,240,192,156,240,192
9008 DATA 48,120,216,220,159,31,15,111
9009 DATA 12,30,27,59,249,248,240,246
9010 DATA x,223,79,111,37,35,1,z
9011 DATA x,251,242,246,164,196,128,z
9012 DATA 3,15,57,3,15,31,47,127
9013 DATA 28,142,131,63,254,248,240,240
9014 DATA 127,47,31,15,3,57,15,3
9015 DATA 240,240,248,254,63,131,142,28
9016 DATA z,z,3,7,15,2,z,z
9017 DATA 7,x,x,159,111,247,240,96
9018 DATA z,z,z,z,z,254,254,254
9019 DATA z,z,z,z,z,x,x,x
9020 DATA 254,254,254,254,254,254,254,254
9021 DATA 254,254,254,4,50,122,122,48
9022 DATA x,x,x,z,z,z,z,z
9023 DATA x,x,x,z,6,15,15,6
9024 DATA 2,2,250,250,254,252,252,248
9025 DATA x,x,x,x,6,15,15,6
9026 DATA 248,252,254,127,184,216,192,128
9027 DATA z,z,7,9,17,17,31,31
9028 DATA 31,31,31,62,61,59,3,1
9029 DATA z,z,z,z,z,63,97,193
9030 DATA z,z,z,z,z,z,128,192
9031 DATA 128,x,x,x,x,254,z,z
9032 DATA 240,254,x,159,111,246,240,96
9033 DATA 1,3,1,z,3,4,14,31
9034 DATA 128,192,192,224,224,112,119,255
9035 DATA 31,63,115,81,169,112,112,32
9036 DATA 254,252,252,234,213,206,14,4
9037 DATA 224,x,x,249,246,239,15,6
9038 DATA z,z,192,224,240,64,z,z
9039 DATA z,z,z,z,z,127,127,127
9040 DATA z,z,z,z,z,x,x,x
9041 DATA 127,127,127,32,76,94,94,12
9042 DATA 127,127,127,127,127,127,127,127
9043 DATA x,x,x,z,96,240,240,96
9044 DATA x,x,x,z,z,z,z,z
9045 DATA x,x,x,x,96,240,240,96
9046 DATA 64,64,95,95,127,63,63,31

```

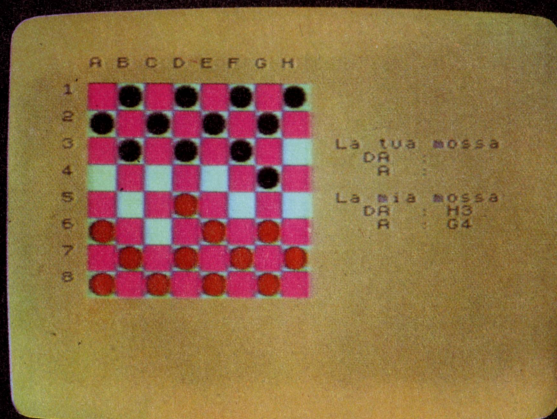

Scacchi



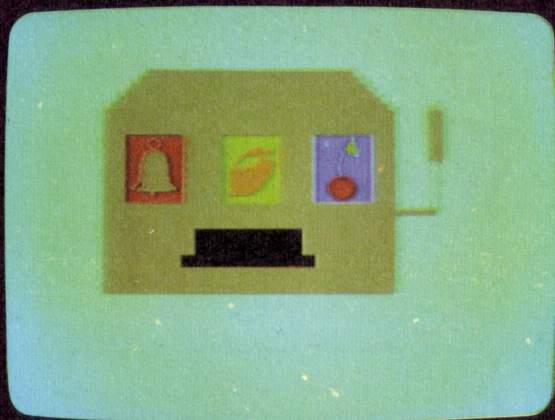
L'avventura

Sei nel soggiorno di una grande casa. Un'insegna dice porta tutti i tesori qui.
 Vedi
 una lampada di ottone
 una porta chiusa
 prendi lampada di ottone
 Sei nel soggiorno di una grande casa. Un'insegna dice porta tutti i tesori qui.
 Vedi
 una porta chiusa
 nord
 Sei in cucina
 Vedi
 un tavolo da cucina
 una bottiglia di rum
 Dimmi cosa devo fare "L"

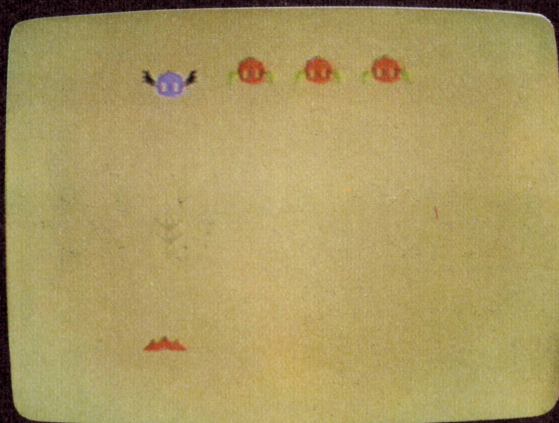
Dama



Fruit Machine



Attacco spaziale




```

9047 DATA z,z,224,144,136,136,248,248
9048 DATA 31,63,127,254,29,27,3,1
9049 DATA 248,248,248,124,188,220,192,128
9050 DATA z,z,z,z,z,z,1,3
9051 DATA z,z,z,z,z,252,134,131
9052 DATA 15,127,x,249,246,111,15,6
9053 DATA 1,x,x,x,x,127,z,z
9054 DATA 1,3,3,7,7,14,238,x
9055 DATA 128,192,128,z,192,32,112,248
9056 DATA 127,63,63,87,171,115,112,32
9057 DATA 248,252,206,138,149,14,14,4

```

FREEWAY FROG

```

20 CLEAR 32303: RANDOMIZE
25 REM LOAD "CODE
26 LET hi=0
30 LET m1=2*INT RND*15: LET m2=2*INT RND*15
33 LET t1=INT RND*30: LET t2=INT RND*30
34 LET c1=2*INT RND*30: LET c2=2*INT RND*30
35 LET ce=2: LET cf=6
36 LET pz=4: LET tz=4: LET mz=2
40 BORDER 0: PAPER 0: INK 7: CLS
45 LET f=4: LET fc=5
50 LET ul=23675: LET uh=23676
55 LET sc=0
60 LET l1=88: LET l2=176: LET l3=48
70 LET h1=127: LET h2=126
75 LET l10=8: LET l11=9: LET l12=7: LET l20=13:
   LET l21=14: LET l22=12
80 PLOT 0,38: DRAW 255,0: DRAW 0,1: DRAW -255,0
90 PLOT 0,128: DRAW 255,0: DRAW 0,1: DRAW -255,0

100 FOR x=2 TO 256 STEP 8: PLOT x,83: DRAW 4,0:
   DRAW 0,1: DRAW -4,0: NEXT x
105 LET pa=20: LET py=20: FOR x=0 TO (fc-1)*3
   STEP 3: LET px=x: GO SUB 1020: NEXT x
106 PRINT AT 0,0; INK 6;"Punteggio "; INK 7;sc;
   AT 0,14; INK 6;"Punteggio max. "; INK 7;hi
110 IF RND<.9 THEN GO TO 500
115 IF RND>.5 THEN GO TO 150
120 LET c=5: FOR j=1 TO 6: BEEP .2,10: BEEP .2,20
   : NEXT j: FOR x=-4 TO 30 STEP 2: GO SUB 1080:
   NEXT x
130 GO TO 500

```

```

150 LET c=5: FOR j=1 TO 6: BEEP .2,20: BEEP .2,10
    : NEXT j: FOR x=30 TO -4 STEP -2: GO SUB 1140
    : NEXT x
500 IF py>15 OR (py<12 AND py>9) OR py<6 THEN
    GO TO 600
510 LET i=22528+py*32+px
520 IF PEEK i<>4 OR PEEK (i+1)<>4 THEN
    GO TO 4000
530 IF PEEK (i+32)<>4 OR PEEK (i+33)<>4
    THEN GO TO 4000
600 IF pz THEN LET pz=pz-1: GO TO 700
610 LET pz=1: LET z$=INKEY$: IF z$="" THEN
    GO TO 700
615 GO SUB 990
620 LET py=py-(z$="q")+(z$="a")
630 LET px=px+(z$="p")-(z$="o")
640 IF z$="q" THEN LET pa=0
641 IF z$="p" THEN LET pa=20
642 IF z$="a" THEN LET pa=40
643 IF z$="o" THEN LET pa=60
650 IF py=0 THEN GO SUB 5000
660 LET sc=sc+10*(z$="q" OR z$="a")
700 GO SUB 1000+pa
750 GO SUB 2100
800 IF mz THEN LET mz=mz-1
810 GO SUB 2000
820 IF NOT mz THEN LET mz=2: GO SUB 2010
900 IF tz THEN LET tz=tz-1
901 GO SUB 2200
902 IF NOT tz THEN GO SUB 2210: LET tz=4
950 GO TO 106
990 PRINT AT py,px;"^.";AT py+1,px;"^.";
991 IF py=10 OR py=11 THEN PLOT px*8+2,83:
    DRAW 4,0: DRAW 0,1: DRAW -4,0: PLOT px*8+10,
    83: DRAW 4,0: DRAW 0,1: DRAW -4,0: RETURN
995 IF py=17 OR py=16 THEN PLOT px*8,38: GO TO 9
    98
996 IF py<>5 AND py<>4 THEN RETURN
997 PLOT px*8,128
998 DRAW 16,0: DRAW 0,1: DRAW -16,0
999 RETURN
1000 POKE u1,13: POKE uh,h2
1010 PRINT INK f;AT py,px;"AB";AT py+1,px;"CD":
    RETURN
1020 POKE u1,13: POKE uh,h2
1030 PRINT INK f;AT py,px;"EF";AT py+1,px;"GH":
    RETURN
1040 POKE u1,13: POKE uh,h2
1050 PRINT INK f;AT py,px;"IJ";AT py+1,px;"KL":
    RETURN

```

```

1060 POKE u1,13: POKE uh,h2
1070 PRINT INK f;AT py,px;"MN";AT py+1,px;"OP":
      RETURN
1080 POKE u1,11: POKE uh,h1
1085 IF x<0 THEN PRINT AT 110,0; INK c;"NO"(
-x TO );AT 111,0;"PQPQAB"(1-x TO ): RETURN
1090 IF x>26 THEN PRINT AT 110,x;"NO"; INK c;"N
O"( TO 30-x);AT 111,x;"NO"; INK c;"PQ"(
TO 30-x): RETURN
1095 PRINT AT 110,x;"NO"; INK c;"NO";AT 111,x;
"NO"; INK c;"PQAB": RETURN
1100 POKE uL,11: POKE uh,h1
1104 IF x>=0 AND x<25 THEN PRINT AT 112,x;"C
DDDD";AT 110,x;"F{4SG8}JK";AT 111,x;"E
EGHHILM": RETURN
1108 IF x<-5 THEN PRINT "";AT 110,0;"JK"(-5-x
TO );AT 111,0;"LM"(-5-x TO ): RETURN
1112 IF x<0 THEN PRINT AT 112,0;"";"CDDDD"(-x
TO );AT 110,0;"F{4SG8}"(-x TO );"JK";AT 11
1,0;"";"EGHHI"(-x TO );"LM": RETURN
1115 IF x=25 THEN PRINT AT 112,x;"CDDDD";;
AT 110,x;"F{4SG8}J";AT 111,x;"EGH
HIL": RETURN
1117 PRINT AT 112,x;"";"CDDDD"( TO 31-x);AT 1
10,x;"";"F{4SG8}"( TO 31-x);AT 111,x;"E
";"EGHHI"( TO 31-x): RETURN
1120 POKE u1,11: POKE uh,h1
1125 IF x<0 THEN PRINT AT 110,0; INK 7;"RS";AT 11
1,0; INK 7;"TU": RETURN
1130 IF x>28 THEN PRINT AT 110,x;"NO";AT 111,x;
"NO": RETURN
1135 PRINT AT 110,x;"RS";AT 111,x;"T
U": RETURN
1140 POKE u1,12: POKE uh,h2
1145 IF x>26 THEN PRINT AT 120,x; INK c;"NO"(
TO 32-x);AT 121,x; INK c;"ABPQ"( TO 32-x):
RETURN
1150 IF x<0 THEN PRINT AT 120,0; INK c;"NO"(
-x TO );AT 121,0; INK c;"BPQ"(-x TO ):
RETURN
1155 PRINT AT 120,x; INK c;"NO";AT 121,x;
INK c;"ABPQ": RETURN
1160 POKE u1,12: POKE uh,h2
1163 IF x>=0 AND x<25 THEN PRINT AT 122,x;"NO
DDDDC ";AT 120,x;"LI{4SG8}E";AT 121,
x;"MKJGGHF": RETURN
1165 IF x=-1 THEN PRINT AT 122,0;"DDDDC";;
AT 120,0;"I{4SG8}E";AT 121,0;"KJGG
HF": RETURN

```

```

1170 IF x<-1 THEN PRINT AT 122,0;"";"▲DDDDC▲"
      (-x TO );AT 120,0;"";"▲{4SG8}E▲"(-x TO );
      AT 121,0;"";"▲JGGHF▲"(-x TO ): RETURN
1173 IF x>29 THEN PRINT AT 120,x;"";"LI"( TO 32
      -x);AT 121,x;"MK"( TO 32-x): RETURN
1175 PRINT AT 122,x;"▲";"DDDDC"( TO 30-x);
      AT 120,x;"LI";"▲{4SG8}E▲"( TO 30-x);AT 121,
      x;"MK";"JGGHF"( TO 30-x): RETURN
1180 POKE u1,12: POKE uh,n2
1185 IF x>29 THEN PRINT AT 120,x; INK 7;"RS";AT 1
      21,x;"TU": RETURN
1190 IF x<0 THEN PRINT AT 120,0; INK 7;"S▲"(-x
      TO );AT 121,0; INK 7;"U▲"(-x TO ):
      RETURN
1195 PRINT AT 120,x;"RS▲";AT 121,x;"TU▲"
      : RETURN
2000 LET c=ce: LET x=c1: GO SUB 1080
2001 LET c=cf: LET x=c2: GO TO 1140
2010 LET c1=c1+2: IF c1=32 THEN LET c1=-4
2011 LET c2=c2-2: IF c2=-6 THEN LET c2=30
2012 RETURN
2100 LET x=m1: GO SUB 1120
2101 LET m1=m1+2: IF m1=32 THEN LET m1=-2
2110 LET x=m2: GO SUB 1180
2111 LET m2=m2-2: IF m2=-4 THEN LET m2=30
2112 RETURN
2200 LET x=t1: GO SUB 1100
2201 LET x=t2: GO TO 1160
2210 LET t1=t1+1: IF t1=32 THEN LET t1=-7
2211 LET t2=t2-1: IF t2=-8 THEN LET t2=31
2212 RETURN
4000 LET f=2: FOR i=50 TO 1 STEP -5: GO SUB 990:
      GO SUB 1000+pa: BEEP .02,i: NEXT i
4005 GO SUB 990: LET f=4
4010 LET fc=fc-1: IF fc<>0 THEN GO TO 105
4020 PRINT AT 1,10;"Game over"
4030 IF sc>hi THEN LET hi=sc: PRINT AT 2,3;"Ma il
      tuo punteggio e' il nuovo▲▲▲▲▲▲PUNTEGGIO MA
      SSIMO"
4040 PRINT AT 4,6;"PREMI 's' per giocare ancora"
4050 IF INKEY$="" THEN GO TO 4050
4060 IF INKEY$="y" THEN GO TO 30
4070 STOP
5000 FOR i=1 TO 50 STEP 5: GO SUB 990: GO SUB 1000
      +pa: BEEP .02,i: NEXT i
5005 GO SUB 990
5010 LET fc=fc-1: IF fc<>0 THEN GO TO 105
5020 GO TO 4020

```

Grafici ad alta risoluzione

Copyright (c) Beam Software

DESCRIZIONE

Questo programma può essere divertente per giocare e vedere il SET di CARATTERI GRAFICI DEFINIBILI DALL'UTENTE enormemente ingrandito.

Ma è ESSENZIALE ai programmatori che desiderano cambiare i CARATTERI GRAFICI in CARATTERI GRAFICI AD ALTA RISOLUZIONE per uso proprio, ad esempio per GIOCHI.

È implementato un'editor eccezionalmente potente con le seguenti caratteristiche:

- Visualizzazione dinamica dello sviluppo grafico
- Immediata cancellazione dei caratteri
- Immediato riempimento dei caratteri
- Display editing ad ampia simulazione
- Immagine specchio dei caratteri esistenti
- Corsore con "wrap – around" in quattro direzioni
- Opzione di uno, due o quattro caratteri
- Colori facilmente identificabili
- Uscita immediata in ogni momento senza cambiamenti

E in più, alla fine dell'editing, le locazioni di memoria di questi caratteri vengono aggiornate automaticamente e tutti i dati corrispondenti ai caratteri vengono visualizzati con l'opzione di mandarli alla stampante.

Sarete meravigliato di quanto può essere versatile questo programma. Quando il computer starà svolgendo il processo, sarete informati di tutto.

Per usare il programma:

EDITING

- controllo cursore
- 5 cursore sinistro
- 6 cursore giù
- 7 cursore su

8 cursore destro
1 accende il PIXEL sotto il cursore
0 spegne il PIXEL sotto il cursore
controllo delle operazioni
b cancella tutti i PIXELS
f riempie tutti i PIXELS
m immagine speculare dei blocchi del carattere
s salva il blocco di caratteri modificati nella memoria
E esce dall'editing senza cambiamenti ai blocchi di caratteri
(NOTA: dovete usare (SHIFT) E)

STRUTTURA DEL PROGRAMMA:

Il programma usa un vettore per simulare i bytes di memoria che definiscono il blocco di caratteri.

Il set di caratteri del primo carattere grafico definito dall'utente incomincia alla locazione di memoria 65368 per macchine di 48K e 32600 per macchine di 16K. Poi seguono 8 bytes per la definizione delle forme di quel carattere.

La struttura del programma è la seguente:

INPUT:

DEFINISCE LE VARIABILI DEL DISPLAY FILE
INSERIRE L'INIZIO DEL BLOCCO DI CARATTERI
INSERIRE IL NUMERO DEI CARATTERI (1 o 2 o 4)
DEFINISCE LE DIMENSIONI DELLA simulazione, puntatore dell'indirizzo di memoria, elaborazione VETTORI dell'immagine speculare

CICLO PRINCIPALE:

Inizializza la simulazione, vettore puntatore indirizzo di memoria
LEGGE I BYTES DI MEMORIA E LI SCRIVE NEL VETTORE DI simulazione
VISUALIZZA FORMA INGRANDITA E REGOLARE DEL BLOCCO DI CARATTERI
ESEGUE LA ROUTINE DI EDITING
FINO A SAVE O EXIT
MESSAGGIO DI OPZIONE DI STAMPA
SE DEVONO ESSERE RIDEFINITI ALTRI CARATTERI VAI A INPUT:

Routine di EDITING

DETERMINA LE DIMENSIONI DEL BLOCCO INGRANDITO PER L'EDITING
E1: IL CURSORE LAMPEGGIA
ACCETTA L'INPUT DI UN TASTO DI CONTROLLO
INTERROMPE IL LAMPEGGIARE DEL CURSORE
CASO: '5' o '8' calcola la nuova posizione del cursore
e vai a E1:

- '0' Pone a 0 il vettore corrente di simulazione, aggiorna lo schermo, visualizza
vai a E1:
- '1' Pone a uno il vettore corrente di simulazione, aggiorna lo schermo, visualizza
vai a E1:
- 'f' o 'b' Riempie l'intero vettore simulazione di 1 se 'f', 0 se 'b'
rivisualizza blocco di caratteri,
vai a E1:
- 's' messaggio dell'opzione di save
se è save mette in memoria
il vettore simulazione se
non è save allora
vai a E1;
- 'm' gira le linee del vettore simulazione e mostra la figura invertita
vai a E1:
- 'E' ritorna.

NOTE SPECIALI

Il lampeggio intermittente del cursore è ottenuto cambiando gli attributi della posizione del carattere nel display file.

La locazione di memoria della posizione del carattere in alto a sinistra nel display file è 22528, perciò

$$3040 \text{ LETa} = 22528 + 32 * x + y; \text{ POKEa, PEEKa} + 128$$

farà lampeggiare la linea x colonna y; allo stesso modo

$$3060 \text{ POKEa, PEEKa} - 128$$

toglierà il lampeggio.

Per facilitare l'hard copy dello schermo sulla stampante un pixel ingrandito viene mostrato con PAPER giallo INK nero; un pixel normale riempito viene mostrato con PAPER bianco INK blu.

Viene usata l'istruzione di Print per mostrare il carattere ingrandito e l'istruzione di Plot per mostrare il carattere regolare.

Per fare riferimento all'esatto carattere grafico definibile dall'utente, l'input di ogni carattere minuscolo sarà trasformato a maiuscolo da

$$8060 \text{ LET K\$} = \text{CHR\$} (\text{CODE K\$} - 32)$$

Viene anche eseguito un test per controllare se il carattere immesso si trova oltre il set di caratteri grafici definibili dall'utente con

$$8070.... \text{ OR} (\text{CODE K\$} + \text{nc} + 79 - 1)) 164$$

cioè il codice ASCII del primo carattere maiuscolo più il numero di caratteri che non dovrebbe eccedere 164, che è il valore ASCII dell'ultimo carattere grafico definibile dall'utente.

Per ottenere l'inizio degli 8 bytes di memoria che definiscono un determinato carattere definibile dall'utente e metterlo nel vettore, si usa

```
8150 LET S( I+1)=USR CHR$( CODE K$ + I)
```

La logica alla base della definizione dell'immagine speculare del vettore sta nel dividere il vettore in due vettori, scambiare i due elementi alle estremità e muoversi all'interno finché viene raggiunto il centro del vettore. In altre parole, una riga di otto elementi sarà divisa in due metà di 4 elementi ciascuno. Il primo elemento della prima metà viene sostituito con l'ultimo elemento della seconda metà (e vice versa). Poi il secondo elemento della prima metà sarà sostituito con il penultimo elemento della seconda metà. Potete trovare questa logica alle linee 3330 - 3400.

GRAFICA AD ALTA RISOLUZIONE

```
100 REM USR
110 REM Caatteri gafici definibili dall'utente
120 GO TO 8000
1000 REM Disegna il set di caratteri
1010 REM Input r,c;coordinate d'inizio
1020 PRINT AT 8,11;"_." : PRINT AT 9,11;"_." : FOR H
    =1 TO nc
1030 LET hr=8*(H=3 OR H=4): LET hc=8*(H=2 OR H=4)
1040 FOR I=1 TO 8
1050 FOR J=1 TO 8
1060 PRINT AT (I+r+hr),(J+c+hc);
1070 IF B(I+hr,J+hc)=0 THEN PRINT PAPER 6; INK 0
    ;"_." : GO TO 1090
1080 PRINT PAPER 7; INK 1;"{SG8}": PLOT 87+J+hc,1
    12-I-hr
1090 NEXT J
1100 NEXT I
1110 NEXT H
1120 LET L$=" "
1130 RETURN
2000 REM
2010 REM caratteri nel vettore
2020 REM Input K$,Output B(I,J)
2030 REM
2040 FOR H=1 TO nc
2050 LET hr=8*(H>2): LET hc=8*(H=2 OR H=4)
2060 FOR I=1 TO 8
```

```

2070 LET m=PEEK (s(H)+I-1)
2080 FOR J=8 TO 1 STEP -1
2090 LET B(I+hr,J+hc)=m-2*INT (m/2)
2100 LET m=INT (m/2)
2110 NEXT J
2120 NEXT I
2130 NEXT H
2140 RETURN
3000 REM Edit dei CARATTERI,x=linea,y=colonna
3010 REM Input B(I,J)
3020 LET hr=8*(nc>2): LET hc=8*(nc=2 OR nc=4)
3030 LET x=r+1: LET y=c+1
3040 LET a=22528+32*x+y: POKE a,PEEK a+128:
    PAUSE 200
3050 IF INKEY$="" THEN GO TO 3050
3060 LET M$=INKEY$: POKE a,PEEK a-128: REM toglie
    il lampeggio dall'ultima posizione
3070 IF M$<"5" OR M$>"8" THEN GO TO 3150
3080 LET y=y+(M$="8" AND (x<>8+hr OR y<>22+hc)
    )-(M$="5" AND (x<>1 OR y<>c+1))
3090 LET x=x+(M$="6")-(M$="7")
3100 IF x>8+hr THEN LET x=1
3110 IF x<1 THEN LET x=8+hr
3120 IF y>22+hc THEN LET y=c+1: LET x=x+1
3130 IF y<15 THEN LET y=22+hc: LET x=x-1
3140 GO TO 3040
3150 IF M$<>"0" AND M$<>"1" THEN GO TO 3190
3160 PRINT AT x,y;
3170 IF B(x-r,y-c)=1 AND M$="0" THEN LET B(x-r,y-
    c)=0: PRINT PAPER 6; INK 0;" ": PLOT OVER 1
    ;87+y-c,112-x+r: GO TO 3040
3180 IF B(x-r,y-c)=0 AND M$="1" THEN LET B(x-r,y-
    c)=1: PRINT PAPER 7; INK 1;"{SG8}": PLOT 87+
    y-c,112-x+r: GO TO 3040
3190 IF M$<>"f" AND M$<>"F" AND M$<>"b"
    AND M$<>"B" THEN GO TO 3240
3200 LET b=(M$="f" OR M$="F")
3210 GO SUB 4000: REM inizializzazione del vettore

3220 GO SUB 1000: REM disegna il carattere
3230 GO TO 3030: REM inizia di nuovo
3240 IF M$<>"E" THEN GO TO 3260
3250 PRINT AT 19,0; INK 2;"Nessun cambiamento per
    questo carattere!": RETURN
3260 IF M$<>"s" AND M$<>"S" THEN GO TO 3320
3270 PRINT AT 18,0;"Lo memorizzi? (s o n) ";
3280 LET e$=INKEY$: IF e$<>"s" AND e$<>"n"
    THEN GO TO 3280
3290 PRINT INK 2;e$: PAUSE 50
3300 IF e$="s" THEN GO SUB 5000: RETURN

```

```

3310 FOR M=1 TO 3: PRINT AT 17+M,0;: PRINT "
                                     ";: NEXT M: GO TO 3
      030
3320 IF M$<>"m" AND M$<>"M" THEN GO TO 3040
3330 PRINT AT 0,24; FLASH 1; INVERSE 1; INK 3;"SPE
      CCHIO": LET fc=INT (bc/2+.5)
3340 FOR I=1 TO br
3350 FOR J=1 TO fc
3360 LET M=B(I,J)
3370 LET B(I,J)=B(I,bc-J+1)
3380 LET B(I,bc-J+1)=M
3390 NEXT J
3400 NEXT I
3410 PRINT AT 0,24;"AAAAAAAAA"
3420 GO SUB 1000
3430 GO TO 3040
4000 REM Inizializza il vettore simulazione
4010 FOR I=1 TO br
4020 FOR J=1 TO bc
4030 LET B(I,J)=b
4040 NEXT J
4050 NEXT I
4060 RETURN
4070 REM
5000 REM Trasporta il vettore in memoria
5010 PRINT AT 0,18; FLASH 1; PAPER 3; INK 7;"MEM
      ORIZZAZIONE"
5020 FOR H=1 TO nc
5030 LET hr=8*(H>2): LET hc=8*(H=2 OR H=4)
5040 FOR I=1 TO 8
5050 LET t=0
5060 FOR J=8 TO 1 STEP -1
5070 IF B(I+hr,J+hc)=1 THEN LET t=t+2↑(8-J)
5080 NEXT J
5090 POKE (s(H)+I-1),t
5100 NEXT I
5110 NEXT H
5120 PRINT AT 0,18;"AAAAAAAAAAAAAAAAA"
5130 PRINT AT 0,19; PAPER 3; INK 7;"MEMORIZZATO!!"

5140 RETURN
6000 REM
6010 REM lista i valori in memoria del vettore
6020 PRINT AT 0,0;"AAAAAAAAA"
6030 FOR H=1 TO nc
6040 LET hc=(H=2 OR H=4)*5+1: LET hr=(H>2)*9
6050 PRINT AT hr,hc-1; PAPER 1; INK 7;"_.";CHR$ (
      CODE K$+H-1);"_"
6060 FOR I=1 TO 8
6070 PRINT AT hr+I,hc;PEEK (S(H)+I-1)

```

```

6080 NEXT I
6090 NEXT H
6100 LET UG=CODE K$+79: PRINT AT 8,11; OVER 0;
CHR$ UG;
6110 IF nc>=2 THEN PRINT CHR$ (UG+1)
6120 IF nc=4 THEN PRINT AT 9,11; OVER 0;CHR$ (U
G+2);CHR$ (UG+3)
6130 RETURN
8000 BORDER 5: INK 0: PAPER 7: OVER 0: FLASH 0:
CLS
8010 DIM K$(1)
8020 PRINT "Quale e' il carattere da ridefi-nire?"
: PRINT "(A - U)"
8030 LET K$=INKEY$: IF K$>="A" AND K$<="U" THEN
GO TO 8060
8040 IF K$<"a" OR K$>"u" THEN GO TO 8030
8050 LET K$=CHR$ (CODE K$-32)
8060 PRINT K$: PRINT : PRINT "Quanti caratteri? (1
- 4) ";
8070 LET nc=((CODE INKEY$)-48): IF nc<1 OR nc>4
OR (CODE K$+nc+79-1)>164 THEN GO TO 8070
8080 IF nc=3 THEN LET nc=4
8090 PRINT nc: FOR I=1 TO 50: NEXT I: CLS
8100 DIM S(nc): REM locazione in memria del set di
caratteri
8110 LET br=8*(1+(nc=4)): LET bc=8*(1+(nc>1))
8120 DIM C(bc): DIM B(br,bc): REM Vettore simulazi
one
8130 FOR I=0 TO nc-1
8140 PRINT INK 2;CHR$ (CODE K$+I);" ";: REM scriv
e i caratteri
8150 LET S(I+1)=USR CHR$ (CODE K$+I): REM locazion
e in memoria del set di caratteri
8160 NEXT I
8170 PRINT AT 0,21; FLASH 1; PAPER 1; INK 7;"CAR
ICAMENTO"
8180 LET b=0: GO SUB 4000: REM inizializza il vett
ore
8190 GO SUB 2000: REM carattere al vettore
8200 LET r=0: LET c=14
8210 GO SUB 1000: REM Riprende il carattere
8220 PRINT AT 0,21;"▲▲▲▲▲▲▲▲▲▲"
8230 GO SUB 3000: REM Edit del carattere
8240 GO SUB 6000: REM lista i valori in memoria
8250 PRINT AT 20,0;"Copia sulla stampante? (s or n
)";
8260 LET g$=INKEY$: IF g$<>"s" AND g$<>"n"
THEN GO TO 8260
8270 PRINT g$
8280 IF g$="n" THEN GO TO 8320

```

```
8290 PRINT AT 21,0;"Premi un tasto quando sei pron  
to": PAUSE 2000  
8300 IF INKEY$="" THEN GO TO 8300  
8310 COPY  
8320 PRINT AT 20,0;L$: PRINT AT 21,0;L$  
8330 PRINT AT 21,0;L$  
8340 PRINT AT 20,0;"Cambi qualche altro carattere?  
 (s or n)";: PAUSE 200  
8350 LET g$=INKEY$: IF g$<>"s" AND g$<>"n"  
 THEN GO TO 8350  
8360 PRINT g$  
8370 IF g$="n" THEN STOP  
8380 CLS : RUN
```

Line Renumbering

Copyright (c) Beam Software

DESCRIZIONE

Queste sono alcune utility che potreste trovare utili se avete l'abitudine di sviluppare programmi e di trovarvi alla fine dello sviluppo con linee numerate in modo bizzarro e senza spazio nel quale inserire quest'ultima routine brillantemente ideata.

Il Modello 1 (RENUM 1) è un breve tipo di renumber che esegue la rinumerazione automatica dalla linea 100, la prima linea, includendo tutte le linee prima della 9990, alla quale inizia il primo programma di renumber. L'incremento delle linee per il modello 1 è 10.

Il Modello 2 è un breve tipo di renumber simile nelle caratteristiche al modello 1 tranne che visualizza sullo schermo tutti i GOTO e GOSUB insieme al nuovo numero di linea.

I GOTO e GOSUB non vengono rinumerati perché lo ZX Spectrum ammette GOTO e GOSUB calcolati e questi sono troppo difficili da rinumerare per il programma. Il meglio che possiamo fare è di segnalare dove sono nella nuova versione e farvi trovare i cambiamenti da fare.

Il Modello 3 (RENUM 3) è la versione lunga del programma di renumber. Vi permette di rinumerare un blocco di linee con ogni valore di incremento visualizzando i GOTO e i GOSUB. Vi permettete anche di stampare i messaggi relativi a questi GOTO e GOSUB sulla stampante. Non solo, questo è un programma che fa largo uso dello schermo e dà molta enfasi alla facilità d'uso.

Per usare questi programmi dovete caricarli con MERGE anziché con LOAD poiché quest'ultimo metodo cancellerebbe il primo programma.

Il formato di una linea BASIC è il seguente:

2-byte: numero di linea con il byte più significativo davanti

2-byte: lunghezza della linea con il byte meno significativo davanti

n byte: contenuto della linea includendo l'ultimo carattere fine linea.

L'indirizzo d'inizio del programma BASIC è 23755.

Modello 1

Usare RUN 9990 oppure GOTO 9990 per far partire il programma.

STRUTTURA DEL PROGRAMMA

```
INIZIALIZZA IL NUOVO NUMERO DI LINEA A 100
INIZIALIZZA IL PUNTATORE DI MEMORIA A 23755
L1: SE IL NUMERO DI LINEA CODIFICATO È > = 9990
    stop
    DECODIFICA IL NUMERO DELLA NUOVA LINEA IN MEMORIA
    CALCOLA IL NUMERO DELLA NUOVA LINEA INCREMENTATA
    MUOVI IL PUNTATORE DI MEMORIA ALLA PROSSIMA LINEA
    VAI A L1:
```

NOTE SPECIALI

Per cancellare il programma di renumber dopo l'uso, potreste aver bisogno di caricare il programma LINE BLOCK DELETE, un'altra interessante UTILITY di questo libro.

Il programma va da 9990 a 9995.

Modello 2

Usare RUN 9988 per far partire il programma.

STRUTTURA DEL PROGRAMMA

```
INIZIALIZZA IL NUOVO NUMERO DI LINEA, L'INCREMENTO, L'INIZIO DEL
BLOCCO, LA FINE DEL BLOCCO
TROVA LA LOCAZIONE DI MEMORIA DELL'INIZIO DEL BLOCCO
L1: SE È RAGGIUNTA LA FINE DEL BLOCCO DI LINEE ALLORA STOP POKE
DELLA NUOVA LINEA NELLA MEMORIA
SCRIVE TUTTI I GOTO E I GOSUB
AGGIORNA IL PUNTATORE DELLA LOCAZIONE DI MEMORIA E IL NUMERO
DI LINEA
VAI A L1:
```

NOTE SPECIALI

Questo modello ha inserito un codice per trovare la locazione di memoria dell'inizio del blocco e potete poi cambiare le variabili alle linee 9988 per adattare alle vostre operazioni.

Ancora, se volete cancellare questo breve programma, potete farlo manualmente o potete usare il nostro LINE BLOCK DELETE, un programma che trovate in questo libro.

Non sono fornite opzioni di stampa per copiare i messaggi dei GOTO e GOSUB; pertanto dovete farlo manualmente.

Il programma va da 9988 a 9999.

Modello 3

Usate RUN 9900 o GOTO 9900 per far partire il programma.

STRUTTURA DEL PROGRAMMA

Vengono usate tecniche di manipolazione dello schermo per l'INPUT dei numeri di linea d'inizio e di fine del blocco da rinumerare, dell'incremento delle linee e della nuova linea d'inizio del blocco.

Tutti i quattro valori inseriti saranno arrotondati all'intero.

La struttura del programma è come segue:

```
INPUT DEI NUMERI DI LINEA DI INIZIO DEL BLOCCO
  INCREMENTA E NUOVO NUMERO D'INIZIO DEL BLOCCO
TROVA L'INIZIO DI MEMORIA DELLA LINEA D'INIZIO
CHIAMA IL MODULO DI RENUMBER (uguale al modello 2)
OPZIONE PER CONTINUARE IL RENUMBER DI ALTRO BLOCCO
  OPPURE PER COPIARE I MESSAGGI DI GOTO E GOSUB
  OPPURE PER USCIRE CON L'OPZIONE DI DELETE DEL MODELLO 3
```

NOTE SPECIALI

Questo modello vi dà l'opportunità di eliminare il modello stesso alla fine dell'operazione.

È incluso il renumber delle linee del blocco.

Il programma va da 9900 a 9999.

Se scegliete l'opzione di eliminare lo stesso modello di renumbering, alla fine della cancellazione, dovrete digitare 9900 (ENTER) per completare la cancellazione. Questo perché tutto il programma di renumber è stato trasformato in una istruzione REM molto lunga.

Se non scegliete l'opzione di cancellazione e alla fine volete cancellare l'intero programma di renumber, potete farlo con GOTO 9945 seguito da (ENTER).

Un uso interessante di questo modello è definire un'istruzione REM contenente il messaggio che volete mostrare, per esempio 10 REM Copyright (c) Beam Software.

Quindi potete rinumerare questa linea a 0 specificando 0 come nuovo numero di linea.

Dopo la rinumerazione non potete più accedere a questa linea 0 a meno che usiate il programma di renumber per cambiare il numero di quella linea a qualsiasi valore maggiore di 0.

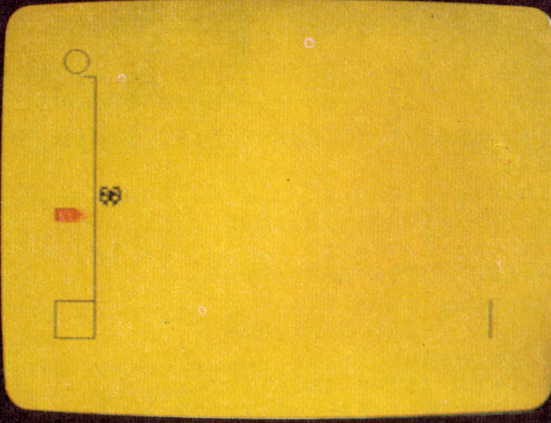
RENUMBER MODELLO 1

```
9990 REM
9991 LET L=100: LET N=23755
9992 IF (256*PEEK N+PEEK (N+1))>=9990 THEN STOP
9993 POKE N,INT (L/256): POKE (N+1),L-INT (L/256)*
  256
9994 LET L=L+10: LET N=N+3+PEEK (N+2)+256*PEEK (N+
  3)+1
```

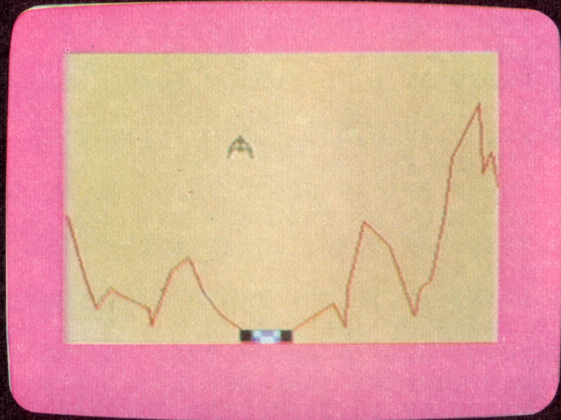
RENUMBER MODELLO 2

```
9988 LET NEW=100: LET INC=10: LET BOB=100: LET EOB
  =9980
9989 LET L=NEW: LET N=23755
9990 IF (256*PEEK N+PEEK (N+1))>=BOB THEN GO TO 9
  992
9991 LET N=N+3+PEEK (N+2)+256*PEEK (N+3)+1:
  GO TO 9990
9992 LET n1=(256*PEEK N+PEEK (N+1)): IF n1>EOB
  THEN STOP
9993 POKE N,INT (L/256)
9994 POKE n+1,L-256*INT (L/256)
9995 FOR I=N+4 TO N+3+PEEK (N+2)+256*PEEK (N+3)
9996 IF PEEK I=236 OR PEEK I=237 THEN PRINT L;"- "
  ;CHR$ PEEK I
9997 NEXT I
9998 LET N=1: LET L=L+INC
9999 GO TO 9992
```

Fuga spaziale



Allunaggio



RENUMBER MODELLO 3

```
9900 REM
9901 REM Programma renumber
9902 LET BOB=100: LET EOB=8999: LET INC=10: LET NE
W=100
9903 BORDER 7: PAPER 7: INK 1: CLS
9904 PRINT AT 0,7; INVERSE 1;"LINE RENUMBER MENU"
9905 PRINT AT 2,17; INVERSE 1;"CORRENTE";AT 2,27;
  INVERSE 1;"NUOVA"
9906 PRINT AT 4,0;"Inizio linea #"
9907 PRINT AT 4,18+4-LEN (STR$ BOB);BOB
9908 PRINT AT 6,0;"Fine^^^linea #"
9909 PRINT AT 6,18+4-LEN (STR$ EOB);EOB
9910 PRINT AT 8,0;"Nuova^^linea #"
9911 PRINT AT 8,18+4-LEN (STR$ NEW);NEW
9912 PRINT AT 10,0;"Incremento"
9913 PRINT AT 10,18+4-LEN (STR$ INC);INC
9914 PRINT AT 14,0; INVERSE 1;"Premi ENTER se non
  vuoi cambiare"
9915 REM inserimento parametri
9916 PRINT AT 4,26; FLASH 1; INVERSE 1;">";
  FLASH 0; INK 2;"^^^"
9917 INPUT S$: IF S$="" THEN GO TO 9919
9918 LET BOB=INT (VAL S$+.5)
9919 PRINT AT 4,26;"^";AT 4,27+4-LEN (STR$ BOB);BO
  B
9920 PRINT AT 6,26; FLASH 1; INVERSE 1;">";
  FLASH 0; INK 2;"^^^"
9921 INPUT S$: IF S$="" THEN GO TO 9923
9922 LET EOB=INT (VAL S$+.5)
9923 PRINT AT 6,26;"^";AT 6,27+4-LEN (STR$ EOB);EO
  B
9924 PRINT AT 8,26; FLASH 1; INVERSE 1;">";
  FLASH 0; INK 2;"^^^"
9925 INPUT S$: IF S$="" THEN GO TO 9927
9926 LET NEW=INT (VAL S$+.5)
9927 PRINT AT 8,26;"^";AT 8,27+4-LEN (STR$ NEW);NE
  W
9928 PRINT AT 10,26; FLASH 1; INVERSE 1;">";
  FLASH 0; INK 2;"^^^"
9929 INPUT S$: IF S$="" THEN GO TO 9931
9930 LET INC=INT (VAL S$+.5)
9931 PRINT AT 10,26;"^";AT 10,27+4-LEN (STR$ INC);
  INC
9932 FOR I=1 TO 50: NEXT I
9933 REM trova il primo numero
9934 LET N=23755
9935 IF 256*PEEK N+PEEK (N+1)>=BOB THEN GO TO 9937
```

```

9936 LET N=(N+3+PEEK (N+2)+256*PEEK (N+3)+1):
      GO TO 9935
9937 CLS : PRINT AT 0,11; FLASH 1; INK 2;
      INVERSE 1;"RINUMERAZIONE"
9938 GO SUB 9991
9939 PRINT AT 0,11; INK 2; INVERSE 1;"...FINITO
      ^"
9940 INPUT """"*"" per uscire,""z"" per copiare lo
      schermo, Qualsiasi tasto per continuare.";S
      $
9941 IF S$="z" THEN COPY : GO TO 9940
9942 IF S$<>"*" THEN GO TO 9903
9943 INPUT """"c"" per cancellare il programma Qual
      siasi tasto per uscire ";k$
9944 IF k$<>"c" THEN STOP
9945 LET N=23755
9946 IF 256*PEEK N+PEEK (N+1)>=9900 THEN GO TO 99
      48
9947 LET N=(N+3+PEEK (N+2)+256*PEEK (N+3)+1):
      GO TO 9946
9948 LET DA=N+2: LET DL=-4
9949 LET n1=PEEK (N+2)+256*PEEK (N+3): LET DL=DL+n
      1+4
9950 LET n2=(256*PEEK N+PEEK (N+1)): IF n2<9999
      THEN LET N=N+3+n1+1: GO TO 9949
9951 LET n}=INT (DL/256): POKE (DA+1),n1: POKE DA,
      DL-n1*256: POKE (DA+2),234
9952 PRINT AT 21,0;"Batti "; FLASH 1;"9900 <ENTER>
      "; FLASH 0;" per cancell."
9953 STOP
9991 LET L=NEW
9992 IF (256*PEEK N+PEEK (N+1))>EOB THEN RETURN
9993 POKE N,INT (L/256)
9994 POKE N+1,L-256*INT (L/256)
9995 FOR I=N+4 TO N+3+PEEK (N+2)+256*PEEK (N+3)
9996 IF PEEK I=236 OR PEEK I=237 THEN PRINT L;"-"
      ;CHR$ PEEK I
9997 NEXT I
9998 LET N=I: LET L=L+INC
9999 GO TO 9992

```


Block Line Delete

Copyright (c) Beam Software

DESCRIZIONE

Questa è un'altra UTILITY che ogni programmatore BASIC dello SPECTRUM troverà utile per lo sviluppo dei programmi.

Se avete letto il programma di RENUMBER in questo libro, noterete che questo BLOCK LINE DELETE è particolarmente utile per cancellare il programma di RENUMBER stesso una volta adempiuta la sua funzione.

Inoltre, questo programma cancellerà anche se stesso.

Di nuovo, vengono usate tecniche di manipolazione dello schermo per inserire le linee iniziali e finali del blocco da cancellare. Il programma è abbastanza abile da cancellare tutte le linee con numeri maggiori o uguali a quello della linea d'inizio del blocco e minori o uguali a quello dell'ultima linea. Così, anche se inserite un numero di linea che non esiste, il programma non si interromperà.

Per usare il programma, prima di tutto dovete inserire i numeri delle linee iniziale e finale del blocco che volete cancellare. Notate che potete cancellare una linea inserendo lo stesso numero per la linea iniziale e finale. Dopo che il programma avrà finito la sua funzione di pulizia, visualizzerà un messaggio che vi chiederà di cancellare il blocco di linee digitando il numero della linea iniziale seguito da (ENTER).

STRUTTURA DEL PROGRAMMA

L'algoritmo usato nel programma conta tutti i bytes della memoria, da dopo la locazione della lunghezza della linea iniziale, al carattere di fine linea della linea finale. Questo valore del byte lunghezza verrà poi inserito (POKE) nella locazione di lunghezza della linea iniziale e il contenuto del primo byte di quella linea sarà modificato a 234 (REM); in effetti, l'intero blocco di linea viene convertito nella singola istruzione REM. Perciò, riinserendo il numero della linea d'inizio senza contenuti, verrà cancellata l'intera linea.

La struttura del programma è come segue:

```
INSERIRE IL NUMERO DELLA LINEA DEL BLOCCO DA CANCELLARE  
MOSTRA MESSAGGIO DI CORREZIONE  
TROVA LA LOCAZIONE DI MEMORIA  
RICORDA LA LOCAZIONE DELLA LUNGHEZZA DELLA LINEA  
E INIZIALIZZA LA NUOVA LUNGHEZZA  
CONTA I BYTE-LUNGHEZZA FINO ALLA FINE DEL BLOCCO  
POKE DEL BYTE-LUNGHEZZA NELLA LOCAZIONE DELLA LUNGHEZZA
```

DELLA LINEA INIZIALE
VISUALIZZA IL MESSAGGIO DI FINE
VISUALIZZA IL MESSAGGIO CON LE ISTRUZIONI FINALI.

NOTE SPECIALI

Dopo che il processo di correzione è finito, se non reinserite il numero di linea iniziale con una stringa nulla, potete ancora far riferimento a ogni linea del blocco tranne la prima. Ogni futuro riferimento alla linea d'inizio cancellerà l'intero blocco.

È opportuno salvare il vostro programma prima di cancellare qualsiasi parte di esso usando questo programma di BLOCK LINE DELETE, nel caso cancelliate il blocco sbagliato.

Dovete caricare questo programma con MERGE; non usate LOAD altrimenti il vostro programma verrà distrutto dal processo di caricamento.

BLOCK LINE DELETE

```
8000 REM
9973 REM
9974 REM BLOCK LINE DELETE
9975 OVER 0: FLASH 0: PAPER 7: INK 1: CLS
9976 PRINT AT 0,8; INVERSE 1;"BLOCK LINE DELETE"
9977 PRINT AT 4,4;"LA CANCELLAZIONE INCLUDE"
9978 PRINT AT 6,7;"DALLA LINEA"
9979 PRINT AT 8,7;"ALLA LINEA";
9980 PRINT AT 6,19; FLASH 1;">"; FLASH 0; INK 2;
    INVERSE 1;"▲▲▲▲"
9981 INPUT S$: IF VAL S$<0 OR VAL S$>9972 THEN
    GO TO 9981
9982 LET SOD=INT (VAL S$)
9983 PRINT AT 6,19;"▲▲▲▲";AT 6,20+4-LEN (STR$ SOD
    );SOD
9984 PRINT AT 8,19; FLASH 1;">"; FLASH 0; INK 2;
    INVERSE 1;"▲▲▲▲"
9985 INPUT S$: IF VAL S$<0 OR VAL S$>9972 THEN
    GO TO 9985
9986 LET EOD=INT (VAL S$)
9987 PRINT AT 8,19;"▲▲▲▲";AT 8,20+4-LEN (STR$ EOD
    );EOD
9988 PRINT AT 11,10; FLASH 1; INK 2; INVERSE 1;"CA
    NCELLAZIONE"
9989 LET N=23755
9990 IF 256*PEEK N+PEEK (N+1)>=SOD THEN GO TO 999
    2
9991 LET N=(N+3+PEEK (N+2)+256*PEEK (N+3)+1):
    GO TO 9990
9992 LET DA=N+2: LET DL=-4
```



```
9993 LET n1=PEEK (N+2)+256*PEEK (N+3): LET DL=DL+n
    1+4
9994 LET n2=(256*PEEK N+PEEK (N+1)): IF n2<EOD
    THEN LET N=N+3+n1+1: GO TO 9993
9995 IF n2>EOD THEN LET DL=DL-n1-4
9996 LET n1=INT (DL/256): POKE (DA+1),n1: POKE DA,
    DL-n1*256: POKE (DA+2),234
9997 PRINT AT 11,10;"^AAAAAAAAAAAAAAAA";AT 11,12;
    INK 2; INVERSE 1;"FINITO"
9998 PRINT AT 15,0;"BATTI "; FLASH 1;SOD;"<ENTER>"
    ; FLASH 0;" PER CANCELLARE";AT 16,0;"IL BLOCC
    O"
9999 STOP
```


Montar la Vostra Macchina

Copyright (c) by Beam Software

In alcuni punti di questo libro abbiamo fatto riferimento a programmi in linguaggio macchina come a programmi più efficienti e, in alcuni casi, vi abbiamo mostrato esempi di come lo scrivere programmi in linguaggio macchina possa ridurre il tempo di esecuzione e anche far risparmiare memoria.

Vi potreste dunque essere chiesti: "Cos'è il linguaggio macchina al quale queste note si riferiscono? Qual è la differenza tra linguaggio macchina e BASIC?"

Per definire correttamente la questione, dovete prima di tutto rendervi conto che lo Spectrum in realtà non esegue mai il vostro programma BASIC. Infatti, il chip che svolge tutto il lavoro, lo Z80, non capisce il BASIC. Tutto ciò che capisce è il suo proprio set di istruzioni.

La gamma di istruzioni che lo Z80 capisce comprende cose semplici come sommare due numeri, sottrarre, confrontare, prendere un numero qui e metterlo là, e così via. La maggior parte delle funzioni più complesse possono essere ottenute con queste semplici operazioni.

Ma persino funzioni leggermente più complesse, semplici per i nostri parametri, come la moltiplicazione e la divisione, non possono essere eseguite come tali.

Dovete scrivere un programma per eseguire queste funzioni.

Il vostro computer Spectrum viene fornito con un programma in linguaggio macchina interno. Questo programma è memorizzato in un chip interno al computer cosicché non deve essere caricato dal nastro ogni qualvolta lo volete usare. Questo programma è nel chip ROM (ROM sta per Read Only Memory, memoria a sola lettura, che significa che non potete modificarla).

La funzione di questo programma è di prendere l'informazione che date al computer via tastiera ed eseguire le istruzioni necessarie a ottenere i risultati che desiderate. Una semplice linea BASIC può richiedere centinaia di linee di programma in linguaggio macchina per eseguire il suo compito.

Potete capire che il BASIC, mentre è un sistema molto più efficiente di programmazione (una linea invece di centinaia) risulta più lento nell'esecuzione dei programmi.

Il programma fornito qui non pretende di insegnarvi il linguaggio macchina; interi libri sono stati dedicati a tale soggetto e, se siete interessati a questo argomento, possiamo raccomandarvi due nostri altri titoli:

PROGRAMMAZIONE DELLO Z80 di R. Zaks.

Z80 PROGRAMMAZIONE IN LINGUAGGIO ASSEMBLY di L.A. Leventhal.

STRUTTURA DEI PROGRAMMI IN LINGUAGGIO MACCHINA

I programmi in linguaggio macchina sono strutturati in modo differente dai programmi BASIC per diversi aspetti. Alcuni di questi sono importanti in questo contesto:

- Non ci sono variabili.
Non c'è equivalente diretto a LET vite = 3
Per come è concepito lo Z80, tutto quello che c'è sono locazioni di memoria.
Se controllate sul manuale del vostro Spectrum troverete alle pagg. 173-176 una lista delle locazioni di memoria usate come variabili dal programma in ROM.
Potete cambiare il valore in quelle locazioni per ottenere diversi effetti e per significare cose differenti, ma questo è il limite dell'analogia alle variabili.
- Non ci sono numeri di linea.
Quando si eseguono le istruzioni, lo Z80 andrà da una istruzione alla successiva, esattamente come le trova.
Tuttavia, vi sono istruzioni che vi consentono di ordinare allo Z80 di saltare ad altre istruzioni o di chiamare un altro set di istruzioni, ma anche questi salti e chiamate vengono specificati in termini di posizioni della memoria — ossia un'istruzione potrebbe essere "vai all'istruzione alla locazione di memoria 16".
Ciò significa che, in generale, i programmi sono progettati per locazioni di memoria ben definite. Alcuni programmi, comunque, sono rilocabili e ne discuteremo più avanti.
- Tutti i numeri usati vanno da 0 a 255 oppure da 0 a 65535, (dipende se occupano una o due posizioni di memoria!).
In alte parole, i numeri che occupano una sola locazione di memoria sono compresi fra 0 e 255. A causa di ciò è stata sviluppata una notazione speciale che descrive efficientemente i numeri nell'intervallo 0 - 255.
Questa viene chiamata notazione esadecimale (o Hex). Non è il caso di spiegare qui questa notazione, ma troverete una tavola di conversione da esadecimale a decimale nel manuale Spectrum alle pagg. 183-188.

CHE COSA SONO PEEK E POKE?

Ora che sapete un po' meglio come funziona lo Spectrum, capite che non ci possono essere equivalenti semplici a Let a = b.

L'equivalente più vicino è pensare ad a e b non come a variabili, ma come contenuti di locazioni differenti. Data questa equivalenza, come è possibile trovare quali sono le diverse locazioni di memoria?

L'istruzione BASIC PEEK (che in Inglese significa "sbirciare") ci permette di fare esattamente quello che suggerisce il suo nome — possiamo applicare PEEK a una locazione di memoria per trovare cosa contiene. Inoltre, PEEK non ne disturba il contenuto. Possiamo fare PEEK alle locazioni desiderate senza arrecare alcun danno o disturbare alcunché.

Così possiamo alla fine trovare cos'è b. Ricordate che guardare in una singola locazione di memoria vi darà in risposta un numero da 0 a 255.

Ora, come possiamo modificare il contenuto di un'altra locazione? Con l'istruzione POKE (in Inglese vuol dire "infilare"). Ancora, questa istruzione ha una funzione che rispecchia fedelmente il suo nome. Prendete il numero che volete mettere nella locazione di memoria, ed eseguite una POKE ad essa. Questa, proprio come suggerisce il nome, è un'istruzione piuttosto rudimentale e definitiva — inserite un valore in un determinato punto e là rimane. Il danno potenziale che potete causare in termini di programmazione è immenso.

Non esitate, tuttavia, a usare questa istruzione e sperimentare con essa. La cosa peggiore che potete fare è bloccare il vostro programma, o dover reinizializzare lo Spectrum. Non potete danneggiare il computer in sé eseguendo una POKE di qualsiasi valore in qualsiasi locazione.

Ricordate, comunque, che potete eseguire solo POKE di valori fra 0 e 255.

NOTAZIONE ESADECIMALE

Potreste aver dato un'occhiata alle pagg. 183-188 del vostro manuale Spectrum ed aver visto la colonna dei numeri esadecimali.

Riguardo a questa notazione, spiccano alcune cose:

- Ogni numero è sempre di 2 caratteri
- Sono ammessi soltanto i numeri da 0 a 9 e le lettere da A a F

Questi due punti sono infatti i maggiori pregi della notazione esadecimale.

Le lettere da A a F rappresentano i numeri da 10 a 15.

Per passare dalla notazione esadecimale a quella decimale prendere il numero nella colonna delle decine, moltiplicarlo per 16 e sommare il numero nella colonna delle unità. Così

0A Hex = 10 decimale

11 Hex = 17 decimale, ecc.

IL PROGRAMMA MONITOR CODICE MACCHINA

Questo programma vi consente di esaminare il contenuto di ogni locazione di memoria dello Spectrum, sia nella ROM che nella RAM (che è la memoria "normale" che include lo schermo, l'area del programma, le variabili, la memoria libera e l'area dei caratteri grafici definibili dall'utente).

Non è il programma più efficace se desiderate inserire molti dati nell'area di memoria, ma è ottimo se desiderate esaminare qualsiasi porzione di memoria e modificarla. Per inserire molti dati nella memoria, fate riferimento ad alcuni dei programmi in questo libro, come il programma DAMA.

Ogni linea del display mostra il contenuto di 10 locazioni di memoria, con il contenuto rappresentato in notazione esadecimale; potete usare le tasti frecce (Shift 6 a 8) per muovere il cursore.

Premendo qualsiasi tasto, che non sia un tasto freccia, potrete cambiare il contenuto di quella locazione di memoria. Il valore che inserite deve essere in formato esadecimale. Ricordate che sono permessi solo i numeri da 0 a 9 e le lettere da A a F e che un numero valido è composto esattamente da due caratteri.

QUALCHE ESERCIZIO INTERESSANTE

- Provate a modificare il contenuto delle locazioni di memoria nella ROM (Indirizzo sotto 16384). Cosa succede?
- Provate a modificare il contenuto delle locazioni di memoria da 22528 a 23295. Questa è l'area degli "attributi", che definisce i colori da mostrare sullo schermo. Non potete danneggiare il programma o il computer.
- Provate a modificare il contenuto delle locazioni di memoria da 16384 a 22527. Questa è l'area di memoria video.
Ricordate che ciascun carattere sullo schermo richiede 8 locazioni nell'area della memoria di schermo per essere definito. Le otto locazioni di memoria non sono consecutive. Potete trovare come queste 8 locazioni di memoria sono in relazione le une alle altre.
Non potete danneggiare il computer o il programma facendo ciò.

PROGRAMMI RILOCABILI IN LINGUAGGIO MACCHINA

Ci sono alcuni programmi in linguaggio macchina che sono detti essere rilocabili, il che significa che non è importante dove si trovino in memoria.

Tali programmi sono ovviamente molto utili quando vengono usati in aggiunta ai programmi BASIC, perché significa che possiamo inserirli ovunque ci sia uno spazio libero.

Un esempio è la routine di movimento dello schermo nel programma ELIMINATOR contenuto in questo libro.

Vi diamo qui un altro utile programma rilocabile in linguaggio macchina, questa volta si tratta di un programma di renumber. Noterete che, se confrontato coi programmi di renumber in BASIC listati altrove in questo libro, la versione in linguaggio macchina risulta molto più corta (solamente 27 bytes!) oltre che enormemente più veloce.

Per caricare il programma in memoria è necessario il listato di un corto programma BASIC, oppure potete inserire i bytes in memoria tramite il programma: "Monitor Codice Macchina". (I valori esadecimali sono nella colonna di sinistra del listato assembly).

Dopo aver inserito i numeri in memoria è una buona idea salvare il programma su nastro, usando l'istruzione SAVE " " CODE.

Per far eseguire un programma in codice macchina viene usata la funzione USR, come PRINT USR 32500 o LET v = USR 32500.

Il numero dopo l'USR deve essere l'indirizzo al quale si trova il programma in codice macchina.

Assicuratevi di porre sempre il vostro programma in codice macchina in un'area che non possa essere sovrascritta, esempio sopra CLEAR o nell'area dei caratteri grafici definibili dall'utente, ecc.

Il programma in codice macchina qui proposto rinumererà tutte le linee, iniziando con la linea 100 e incrementando con passo 10. Potete cambiare il numero d'inizio modificando il 5° numero (attualmente 90 = 100 - 10) e l'incremento modificando il 13° numero (attualmente 10). Notate che NON vengono rinumerati i GOTO e i GO-SUB.

Listato assembly del programma di renumber in codice macchina

| | | |
|--------------|--------------|-------------------------------------|
| 11 CA 5C | LD DE, 5C0AH | ;inizio del BASIC |
| 21 5A 00 | LD HL, 90 | ;numero inizio - passo |
| 13 nxline | INC DE | |
| 1A | LD A, (DE) | ;che cos'è? |
| FE 28 | CP 28H | ;fine linea? |
| DO | RET NC | ;se no, finito! |
| 01 0A 00 | LD BC, 10 | ;dimensione passo |
| 09 | ADD HL, BC | ;numero nuova linea |
| EB | EX DE, HL | ;sostituzione temporanea |
| 72 | LD (HL), D | ;inserisci il nuovo numero di linea |
| 23 | INC HL | |
| 73 | LD (HL), E | |
| 23 | INC HL | ;prendi lunghezza linea |
| 4E | LD C, (HL) | ;mettila in BC |
| 23 | INC HL | |
| 46 | LD B, HL | |
| 09 | ADD HL, BC | ;posizione di fine linea |
| EB | EX DE, HL | ;fine sostituzione |
| 18 EB | JR nxtlin | ;fai prossima linea |

MONITOR CODICE MACCHINA

```
100 PRINT INK 7; PAPER 2; AT 0,0;"MONITOR.CODICE.
    MACCHINA...."
110 INPUT "Inserire l'indirizzo di inizio... (in d
    ecimale) ";s
120 LET s=10*INT (s/10)
130 LET x=0: LET y=0
140 GO SUB 1000
150 PRINT INK 7; PAPER 2; AT 0,0;"MONITOR CODICE.
    MACCHINA...."
160 PRINT OVER 1; FLASH 1; AT 3+2*x,3+3*y;" "
170 LET a$=INKEY$
180 IF CODE a$<12 AND CODE a$>7 THEN GO TO 600
190 IF CODE a$=0 THEN GO TO 170
200 INPUT "CAMBI IL VALORE DEL BYTE A (Hex)";a$
210 IF LEN a$<>2 THEN GO TO 160
220 GO SUB 500
230 IF v<0 OR v>15 THEN GO TO 170
240 LET h=v
250 LET a$=a$(2)
260 GO SUB 500
270 IF v<0 OR v>15 THEN GO TO 170
280 LET l=v
290 LET v=16*h+1
300 POKE s+10*x+y,v
310 PRINT OVER 1; FLASH 0; AT 3+2*x,3+3*y;" "
320 IF y=9 THEN GO SUB 1000
330 LET y=y+1
340 IF y>9 THEN LET y=0: LET x=x+1
350 IF x<10 THEN GO TO 140: REM altrimenti cade
    nella pagina successiva
360 GO TO 730
500 LET v=CODE a$-48-7*(CODE a$>64)-32*(CODE a$>96)
510 RETURN
600 PRINT OVER 1; FLASH 0; AT 3+2*x,3+3*y;" "
610 GO TO 540+10*CODE a$
620 LET y=y-1: GO TO 660
630 LET y=y+1: GO TO 700
640 LET x=x+1: GO TO 730
650 LET x=x-1: GO TO 680
660 IF y>=0 THEN GO TO 160
670 LET y=9: LET x=x-1
680 IF x>=0 THEN GO TO 140
690 LET s=s-10: CLS : GO TO 130
700 IF y<10 THEN GO TO 160
710 LET y=0: LET x=x+1
720 IF x<10 THEN GO TO 140: REM altrimenti cade
    nella pagina successiva
```



```

730 IF x=10 THEN CLS : LET s=s+10*x: GO TO 130
740 GO TO 140
1000 IF s<0 OR s>65530 THEN GO TO 110
1010 PRINT AT 2+2*x,0;s+10*x: PRINT TAB 2;
1020 FOR i=s+10*x TO s+10*x+9
1030 IF i>65535 THEN LET i=s+10*1+9: GO TO 150
1040 LET v=PEEK i: LET h=INT (v/16): LET l=v-16*h
1050 PRINT "."+CHR$ (h+48+7*(h>9))+CHR$ (l+48+7*(l
>9));
1060 NEXT i
1070 RETURN

```

RENUNBER IN CODICE MACCHINA

```

100 CLEAR 32500: LET a=32500
110 READ n: POKE a,n
120 LET a=a+1: GO TO 110
130 DATA 17,202,92,33,90,0,19,26,254,40,208,1,10,
0,9,235,114,35,115,35,78,35,70,9,235,24,235
140 FOR i=0 TO 26
150 PRINT PEEK (32500+i);".";
160 NEXT i

```


Libro paga

Copyright (c) Beam Software

DESCRIZIONE

Il foglio elettronico di questo libro paga calcolerà il reddito dei dipendenti e produrrà un registro che può essere usato per calcolare le paghe.

Il registro inizia con l'opzione

- O Inserire il nome del dipendente e la sua paga oraria, il coefficiente di lavoro straordinario. Viene fornita un'opzione per salvare su nastro questi dati sui dipendenti.
- O Potete riprendere i dati sui dipendenti che avete in precedenza salvato su nastro.

Alla fine di ciascun periodo di paga, inserite le ore di ogni dipendente (normali e straordinarie). Il programma calcolerà le entrate lorde di tutti i dipendenti.

Questo programma può provvedere soltanto ad un massimo di 18 dipendenti. Ma per una piccola impresa questo farà sicuramente risparmiare, a voi o alla vostra segreteria, molto tempo nel calcolare i salari.

L'intero programma fa uso dello schermo, con un appropriato cursore lampeggiante che indica quale campo il programma si aspetta di ricevere. Tutti gli input e gli output di dati numerici saranno giustificati a destra.

Alla fine degli inserimenti e dei calcoli la pressione di ogni tasto vi porterà al prossimo dei tre principali schermi del programma nel seguente ordine:

Dati sui dipendenti
Registro ore del libro paga
Registro salari del libro paga.

Il calcolo del salario viene arrotondato alla lira più vicina.

STRUTTURA DEL PROGRAMMA

Questo programma è uno dei tre modelli finanziari per lo SPECTRUM di questo libro. Una delle maggiori differenze fra questo tipo di programma e i giochi o le utilities è che gli INPUT e gli OUTPUT devono essere fatti in modo semplice. In termine di procedura noterete che si tratta principalmente di manipolazioni di stringhe e di calcoli.

La struttura del programma è la seguente:

INIZIALIZZAZIONE

INIZIALIZZA IL MODO INSERIMENTO DEI TRE SCHERMI
INIZIALIZZA LE VARIABILI DEL VIDEO

DATI SUI DIPENDENTI

SE 'E' o 'e' (entrata nuova informazione)
inserire numero di dipendenti (da 1 a 18)
definisce le dimensioni dei seguenti vettori
E\$ Vettore dati dipendente
R\$ Vettore ore normali
O\$ Vettore ore straordinarie
G\$ Vettore entrate lorde

SE 'R' o 'r' (richiama le informazioni)
carica da nastro i dati degli impiegati
definisce le dimensioni dei vettori di cui sopra tranne E\$

I: DEFINISCE LO SCHERMO CON I DATI DEI DIPENDENTI
SE IL MODO RM È INSERIMENTO (rm=0)
poni rm = 1
inserisci nomi dipendenti e paga oraria
inserisci coefficienti straordinari
opzione salvataggio dati dipendenti
ritorna

RISCRIVI I DATI DEI DIPENDENTI
opzione per mandare il video alla stampante
ritorno

REGISTRO ORE DEL LIBRO PAGA

DEFINISCI LO SCHERMO DEL REGISTRO ORE DEL LIBRO PAGA
SE IL MODO PM È INPUT (pm = 0)
poni il modo PM per visualizzare (pm = 1)
inserisci le ore normali e le ore straordinarie
ritorna

RISCRIVI I MESSAGGI DELLE ORE SUL LIBRO PAGA
opzione per mandare il video allo stampato
ritorna

REGISTRO SALARI LORDI DEL LIBRO PAGA

SE IL MODO PR È CALCOLO (pr = 0)
definisci modo display (pr = 1)
inizializza TP (salari lordi totali)

calcola e somma tutti i salari dei dipendenti
e inseriscili nei vettori corrispondenti
DEFINISCI I MESSAGGI DEI SALARI DEL LIBRO PAGA
RISCRIVI I MESSAGGI DEI SALARI DEL LIBRO PAGA
opzione per mandare il video alla stampante

CICLO PRINCIPALE

DATI SUI DIPENDENTI

L1: Registro ore del LIBRO PAGA

Registro delle entrate del LIBRO PAGA

GOSUB L1:

GOTO L1:

NOTE SPECIALI

Il vettore dei dati dei dipendenti è strutturato come segue:

— primo record

1 - 5 numero dei records dei dipendenti nel vettore

16 - 21 coefficienti straordinari

— records seguenti

1 - 15 nome del dipendente

16 - 21 paga oraria del dipendente

Noterete che vi sono tre vettori principali, ciascuno avente principalmente due compiti da eseguire, vale a dire, visualizzazione e inserimento calcoli.

Per scegliere quale è il compito da eseguire, il programma prima definisce tre variabili di modo per inserimento/calcoli (cioè 0) all'inizio e questi a turno verranno messi al modo visualizzazione (cioè 1) dopo che ogni routine sia passata attraverso il suo compito di inserimento/calcolo.

D'ora in poi, ulteriori inserimenti nel vettore daranno luogo all'esecuzione della visualizzazione.

La routine 3000 esegue il compito di convertire ogni stringa numerica inserita col numero di cifre prima del punto decimale e due cifre dopo il punto decimale.

LIBRO PAGA

```
100 REM LIBRO PAGA
110 REM Programma mini libro paga
120 LET rm=0: LET pm=rm: LET pr=rm: REM Definisce
    il modo inserimento
130 OVER 0: PAPER 7: INK 1: FLASH 0: CLS :
    GO TO 9000
1000 REM
1010 REM Routine inizializzazione
1020 CLS : PRINT AT 0,5; INVERSE 1; INK 1;"INFORMA
    ZIONI DIPENDENTI"
1030 PRINT AT 6,0; INK 1;"INSERISCI nuove informaz
    ioni - I"
1040 PRINT AT 8,0; INK 1;"RIPRENDI le informazioni
    ^----- R"
1050 PRINT AT 15,8; INK 1;"Premi".""; FLASH 1;"I";
    FLASH 0;"^o.^"; FLASH 1;"R"; FLASH 0;"^"
    "
1060 LET k$=INKEY$: IF k$<>"I" AND k$<>"R"
    AND k$<>"i" AND k$<>"r" THEN GO TO 1060
1070 IF k$="i" OR k$="r" THEN LET k$=CHR$(CODE k
    $-32)
1080 PRINT AT 15,15; INK 1; INVERSE 1;"I";AT 15,22
    ;"R": PRINT AT 18,12;
1090 IF k$<>"R" THEN GO TO 1150
1100 PRINT INVERSE 1; INK 2;"RICHIAMO";: LET rm=1
1110 INPUT "Premi <spazio><ENTER> quando sei pront
    o ";k$
1120 IF k$<>"^" THEN GO TO 1110
1130 LOAD "minipay" DATA E$( )
1140 LET NE=VAL E$(1,1 TO 2): LET F=VAL E$(1,15
    TO 21): GO TO 1190
1150 PRINT INVERSE 1; INK 2;" INSERISCI ":
    PRINT AT 21,2; INK 1; INVERSE 1; FLASH 1;">"
    ; FLASH 0; INVERSE 0;"no. di dipendenti (1^--
    18):"; INVERSE 1;" "
1160 INPUT NE: IF NE>18 OR NE<1 THEN GO TO 1160
1170 PRINT AT 21,30;"^.";CHR$(8);CHR$(8); INK 1;
    INVERSE 1;NE
1180 DIM E$(NE+1,21)
1190 DIM R$(NE,6)
1200 DIM O$(NE,6)
1210 DIM G$(NE,8)
1220 FOR I=1 TO 75: NEXT I
1230 INK 1: PAPER 7: CLS : PRINT AT 0,2; INVERSE 1
    ;"INFORMAZIONI SUI DIPENDENTI"
```

```

1240 PRINT AT 2,2; INVERSE 1;"NOME DIPENDENTE";
      AT 2,21;"PAGA ORARIA"
1250 IF rm=1 THEN GO TO 1500
1260 REM Inserimento informazioni sui dipendenti
1270 LET rm=1
1280 FOR I=2 TO NE+1
1290 LET E$(I,1 TO 15)=" ": LET E$(I,16 TO 21)="00
      0000"
1300 LET ln=I+3-2
1310 PRINT AT ln,1;; IF I>10 THEN PRINT AT ln,0;
1320 PRINT I-1; FLASH 1; INVERSE 1;">";AT ln,3;
      FLASH 0; INK 2;"▲▲▲▲▲▲▲▲▲▲"
1330 INPUT S$: IF S$="*" THEN RETURN
1340 LET E$(I,1 TO 15)=S$: PRINT AT ln,2;" ";E$(I,
      1 TO 15)
1350 PRINT AT ln,20; INVERSE 1; FLASH 1;">";AT ln,
      23; FLASH 0; INK 2;"▲▲▲▲▲"
1360 INPUT S$: IF VAL S$<0 THEN GO TO 1360
1370 LET ns=3: GO SUB 3000: REM input stringa nume
      rica
1380 PRINT AT ln,20;" ";AT ln,23;"▲▲▲▲ ";AT ln,23
      ;S$: LET E$(I,16 TO 21)=S$
1390 NEXT I
1400 PRINT AT 21,0; INVERSE 1;"Coefficiente straord
      inari";AT 21,25; FLASH 1;">"; FLASH 0; INK 2;
      AT 21,27;"▲▲▲▲▲"
1410 INPUT S$: IF VAL S$<0 THEN GO TO 1410
1420 LET F=VAL S$: PRINT AT 21,25;"▲▲▲▲▲";AT 21,
      25;F: LET E$(1,16 TO 21)=S$
1430 LET E$(1,1 TO 2)=STR$ NE
1440 FOR I=1 TO 50: NEXT I
1450 INPUT "SAVE dei DATI dei DIPENDENTI ? (si o n
      o)";k$
1460 IF k$="" THEN GO TO 1450
1470 IF k$(1)="s" OR k$="S" THEN SAVE "minipay"
      DATA E$(): RETURN
1480 IF k$(1)<>"n" AND k$(1)<>"N" THEN
      GO TO 1450
1490 RETURN
1500 REM Rivisualizza i records degli impiegati
1510 LET NE=VAL (E$(1,1 TO 2)): LET F=VAL (E$(1,16
      TO 21))
1520 FOR I=2 TO NE+1
1530 LET ln=I+3-2
1540 PRINT AT ln,2;E$(I,1 TO 15);AT ln,23;E$(I,16
      TO 21)
1550 NEXT I
1560 PRINT AT 21,0; INVERSE 1;"COEFFICIENTE STRAOR
      DINARI"
1570 PRINT AT 21,25;F

```

```

1580 LET k$=INKEY$: IF k$="" THEN GO TO 1580
1590 IF k$="z" THEN COPY : GO TO 1580
1600 RETURN
2000 REM
2010 REM Inserimento/visualizzazione ore del libro
    paga
2020 CLS : PRINT AT 0,8; INVERSE 1;"REGISTRO LIBRO
    PAGA"
2030 PRINT AT 1,17; INVERSE 1;"---- ORE ----"
2040 PRINT AT 2,0; INVERSE 1;"DIPENDENTE";AT 2,19;
    "REG";AT 2,27;"ST"
2050 IF pm=1 THEN GO TO 2500
2060 LET pm=1: LET TR=0: LET TO=TR
2070 FOR I=2 TO NE+1
2080 LET ln=I+3-2
2090 PRINT AT ln,0;E$(I,1 TO 15)
2100 PRINT AT ln,16; FLASH 1;">"; FLASH 0;
    INVERSE 1; INK 2;"^^^^^^"
2110 INPUT n1: IF n1<0 THEN GO TO 2110
2120 LET S$=STR$ n1: LET ns=3: GO SUB 3000
2130 PRINT AT ln,16;" ";S$: LET TR=TR+n1: LET R$(I
    -1)=S$
2140 PRINT AT ln,24; FLASH 1;">"; FLASH 0; INK 2
    ; INVERSE 1;"^^^^^^"
2150 INPUT n1: IF n1<0 THEN GO TO 2150
2160 LET S$=STR$ n1: LET ns=3: GO SUB 3000
2170 PRINT AT ln,24;" ";S$
2180 LET TO=TO+n1: LET O$(I-1)=S$
2190 NEXT I
2200 LET S$=STR$ TR: LET ns=4: GO SUB 3000
2210 PRINT AT 21,8; INVERSE 1;"TOTALI";AT 21,16;
    INVERSE 0;S$
2220 LET S$=STR$ TO: GO SUB 3000
2230 PRINT AT 21,24;S$
2240 RETURN
2500 REM Visualizza messaggi orari
2510 FOR I=1 TO NE
2520 LET ln=I+3-1
2530 PRINT AT ln,0;E$(I+1,1 TO 15);AT ln,17;R$(I,1
    TO 6);AT ln,25;O$(I,1 TO 6)
2540 NEXT I
2550 PRINT AT 21,8; INVERSE 1;"TOTALI"
2560 LET S$=STR$ TR: LET ns=4: GO SUB 3000
2570 PRINT AT 21,16;S$
2580 LET S$=STR$ TO: GO SUB 3000
2590 PRINT AT 21,24;S$
2600 LET k$=INKEY$: IF k$="" THEN GO TO 2600
2610 IF k$="z" THEN COPY : GO TO 2600
2620 RETURN
3000 REM

```



```

3010 REM input stringa di valori
3020 LET T$=STR$ (INT VAL S$)
3030 FOR J=1 TO (ns-LEN T$): LET T$="."+T$: NEXT J
3040 LET U$=STR$ (INT ((VAL S$-VAL T$)*100+.5)):
    LET T$=T$+"."
3050 IF VAL U$=0 THEN GO TO 3090
3060 FOR J=1 TO LEN U$: IF U$(J TO J)="." THEN
    GO TO 3080
3070 NEXT J
3080 LET U$=U$(1 TO (J-1))
3090 IF VAL U$<10 THEN LET U$="0"+U$
3100 LET S$=T$+U$
3110 RETURN
4000 REM
4010 REM calcola/visualizza le paghe
4020 IF pr=1 THEN GO TO 4100
4030 LET pr=1: LET TP=0
4040 FOR I=1 TO NE
4050 LET n1=(VAL R$(I)+VAL O$(I)*F)*(VAL E$(I+1,16
    TO 21))
4060 LET S$=STR$ n1: LET ns=5: GO SUB 3000
4070 LET G$(I,1 TO 8)=S$
4080 LET TP=TP+n1
4090 NEXT I
4100 CLS : PRINT AT 0,8; INVERSE 1;"REGISTRO LIBRO
    PAGA"
4110 PRINT AT 2,0; INVERSE 1;"IMPIEGATO";AT 2,21;"
    PAGA LORDA";AT 3,19;"£"
4120 FOR I=1 TO NE
4130 LET ln=I+3-1
4140 PRINT AT ln,0;E$(I+1,1 TO 15);AT ln,22;G$(I,1
    TO 8)
4150 NEXT I
4160 PRINT AT 21,3; INVERSE 1;"STIPENDI TOTALI";
    AT 21,19;"£"; INVERSE 0;
4170 LET S$=STR$ TP: LET ns=6: GO SUB 3000
4180 PRINT "."+S$
4190 LET k$=INKEY$: IF k$="" THEN GO TO 4190
4200 IF k$="z" THEN COPY : GO TO 4190
4210 RETURN
9000 REM
9010 REM Routine principale
9020 GO SUB 1000: REM inizializzazione
9030 GO SUB 2000: REM registro libro paga
9040 GO SUB 4000: REM visualizza/calcola la paga
9050 GO SUB 1230: REM record libro paga
9060 GO TO 9030

```


Analisi delle vendite

Copyright (c) by Beam Software

Questo programma usa la storia di vendita di una società per un certo numero di anni per determinare le oscillazioni stagionali e quali sono state le tendenze complessive per quel numero di anni.

I risultati di questo programma sarebbero utili a ogni società e in particolare ai direttori delle vendite di società i cui affari siano affetti da oscillazioni stagionali. Per esempio nell'editoria, circa il 25% di vendite annuali si hanno in novembre e dicembre.

L'informazione richiesta dal programma è il valore delle vendite di ciascun trimestre per gli anni di cui i dati sono a disposizione. Così com'è, il programma può accettare informazioni per un massimo di 18 anni.

Dopo aver inserito queste informazioni, il programma calcolerà i rapporti stagionali dividendo le vendite effettive per la media delle vendite trimestrali per tutti gli anni.

La media dei rapporti di ogni trimestre rispetto gli anni produce l'indice stagionale.

Il programma tratterà anche in forma grafica le vendite annuali della società, fornendo un'informazione visiva immediata del suo sviluppo.

NOTE AL PROGRAMMA:

Come con gli altri programmi d'affari presentati in questo libro, è stata enfatizzata l'interazione tra l'utente e il computer. Normalmente i programmi d'affari devono essere in grado di essere capiti ed essere utili a persone che non hanno conoscenze di computer.

Inserimento di dati pilotati dallo schermo, è il nome dato all'INPUT/OUTPUT usati in questo programma. Il principale vantaggio di ciò è che l'utente ha una verifica immediata di ciò che ha inserito e della quantità di informazioni inserite.

I risultati sono visualizzati su due schermi — uno contenente le vendite effettive e l'altra gli indici stagionali.

Premendo qualsiasi tasto potete passare da uno schermo all'altro. Un'ulteriore opzione vi permette di mandare lo schermo alla stampante.

Il display grafico fa uso delle funzioni PLOT e DRAW dello Spectrum. Viene usato lo schermo della storia delle vendite, lasciando sullo schermo solo le vendite annuali e la media trimestrale. Usando le vendite minime e massime di tutti gli anni per definire le scale degli assi, il grafico fa uso dell'intera area video disponibile.

ANALISI DELLE VENDITE

```
100 REM
110 REM INDICE STAGIONALE
120 BORDER 7: PAPER 7: INK 1: FLASH 0: OVER 0:
    CLS
130 GO TO 8000
1000 REM
1010 REM input anni in esame
1020 PRINT AT 0,9; INVERSE 1;"INDICE STAGIONALE"
1030 PRINT AT 4,0;"STORIA DELLE VENDITE (inclusi)"

1040 PRINT AT 7,6;"DALL'ANNO"
1050 PRINT AT 10,6;"ALL'ANNO"
1060 PRINT AT 7,17; FLASH 1; INVERSE 1;">";
    FLASH 0; INK 2; INVERSE 1;"▲▲▲▲"
1070 INPUT S$: LET y1=INT (VAL S$): IF y1<1900
    OR LEN (STR$ y1)>4 THEN GO TO 1070
1080 PRINT AT 7,17;"▲";y1
1090 PRINT AT 10,17; FLASH 1; INVERSE 1;">";
    FLASH 0; INK 2; INVERSE 1;"▲▲▲▲"
1100 INPUT S$: LET y2=INT (VAL S$): IF y2<1900
    OR LEN (STR$ y2)>4 OR y2<y1 OR y2>y1+17
    THEN GO TO 1100
1110 PRINT AT 10,17;"▲";y2
1120 LET NY=y2-y1+1
1130 DIM Y$(NY,4): REM anno
1140 DIM H$(NY,4,4): REM storia delle vendite
1150 DIM A$(NY,7): REM vendite annuali medie
1160 DIM R$(NY,4,6): REM tassi
1170 DIM D$(4,6): REM indici stagionali
1172 DIM Q(4): REM somma degli indici trimestrali
1174 DIM C(NY): REM disegna la curva
1180 DIM C(NY): REM disegna la curva
1190 DIM C(NY): REM disegna la curva
1200 FOR I=y1 TO y2: LET Y$(I-y1+1)=STR$ I: NEXT I

1210 RETURN
2000 REM
2010 REM input storia vendite
2020 CLS : PRINT AT 0,10; INVERSE 1;"STORIA DELLE
    VENDITE"
2030 PRINT AT 1,26; INVERSE 1;"M.";AT 2,0;
    INVERSE 1;"ANNO";AT 2,5; INVERSE 1;"TRI1";
    AT 2,10;"TRI2";AT 2,15; INVERSE 1;"TRI3";AT 2
    ,20; INVERSE 1;"TRI4";AT 2,26; INVERSE 1;"VEN
    DITE"
2040 IF hm=1 THEN GO TO 2500
2050 FOR I=1 TO NY
```

```

2060 LET ln=I+3-1: PRINT AT ln,0;Y$(I)
2070 LET hp=4: LET ts=0
2080 FOR J=1 TO 4
2090 PRINT AT ln,hp; FLASH 1;">"; INVERSE 1;
      FLASH 0; INK 2;"▲▲▲▲"
2100 INPUT S$: IF VAL S$<0 THEN GO TO 2100
2110 LET n1=INT (VAL S$+.5): LET ts=ts+n1: LET S$=
      STR$ n1
2120 LET Z$="▲▲▲▲": LET Z$(5-LEN S$ TO )=S$
2130 PRINT AT ln,hp;"▲";Z$
2140 LET H$(I,J)=Z$: LET hp=hp+5
2150 NEXT J
2160 LET S$=STR$ (ts/4): LET ns=4: LET nd=2
2170 GO SUB 7000
2180 PRINT AT ln,hp+1;S$
2190 LET A$(I)=S$
2200 NEXT I
2210 LET hm=1: GO TO 2550
2500 REM
2510 REM visualizza
2520 FOR I=1 TO NY
2530 PRINT AT I+3-1,0;Y$(I);"▲";H$(I,1);"▲";H$(I,2
      );"▲";H$(I,3);"▲";H$(I,4);"▲";A$(I)
2540 NEXT I
2550 INPUT "Lo copio su stampante? (s o n)";k$
2560 IF k$="s" OR k$="S" THEN COPY : GO TO 2550
2570 IF k$="N" OR k$="n" THEN RETURN
2580 IF k$="E" THEN CLEAR : STOP
2590 GO TO 2550
3000 REM
3010 REM calcola i coefficienti
3020 CLS : PRINT AT 0,9; INVERSE 1;"COEFFICIENTI CA
      LCOLATI"
3030 PRINT AT 2,0; INVERSE 1;"ANNO";AT 2,6;"TRI1";
      AT 2,13;"TRI2";AT 2,20;"TRI3";AT 2,27;"TRI4"
3040 PRINT AT 21,0; INVERSE 1;"INDEX"
3050 IF cm=1 THEN GO TO 3500
3060 PRINT AT 1,11; FLASH 1; INK 3;"CALCOLO"
3070 FOR I=1 TO 4: LET Q(I)=0: NEXT I
3080 FOR I=1 TO NY
3090 LET n1=VAL A$(I)
3100 FOR J=1 TO 4
3110 LET n2=VAL H$(I,J): LET n3=n2/n1
3120 LET S$=STR$ n3: LET ns=1: LET nd=4
3130 GO SUB 7000
3140 LET R$(I,J)=S$: LET Q(J)=Q(J)+VAL S$
3150 BEEP .03,12
3160 NEXT J
3170 NEXT I
3180 FOR I=1 TO 4

```

```

3190 LET n1=Q(I)/NY
3200 LET S$=STR$ n1: LET ns=1: LET nd=4
3210 GO SUB 7000
3220 LET D$(I)=S$
3230 BEEP .05,24
3240 NEXT I
3250 PRINT AT 1,11;"^"; INVERSE 1; INK 3;"FINITO";
      INVERSE 0; INK 1;"^^"
3260 LET cm=1
3270 NEXT I
3280 PRINT AT 1,11;"^"; INVERSE 1; INK 3;"FINITO";
      INVERSE 0; INK 1;"^^"
3290 LET cm=1
3500 REM
3510 REM Rivisualizza coefficienti/indici
3520 FOR I=1 TO NY
3530 PRINT AT I+3-1,0;Y$(I);"^";R$(I,1);"^";R$(I,2
);"^";R$(I,3);"^";R$(I,4)
3540 NEXT I
3560 PRINT AT 21,5;D$(1);"^";D$(2);"^";D$(3);"^";D
$(4)
3570 INPUT "Lo copio su stampante? (s o n)";k$
3580 IF k$="s" OR k$="S" THEN STAMP COPY : GO TO 3570
3590 IF k$="n" OR k$="N" THEN RETURN
3600 IF k$="E" THEN CLEAR : STOP
3610 GO TO 3570
4000 REM
4010 REM plot grafico vendite
4020 CLS : PRINT AT 0,10; INVERSE 1;"GRAFICO VENDI
TE"
4030 PRINT AT 2,0; INVERSE 1;"ANNO";AT 1,26;
      INVERSE 1;"MED";AT 2,26; INVERSE 1;"VENDIT
E"
4040 IF gm=1 THEN GO TO 4500
4050 LET gm=1
4060 PRINT AT 2,12; INVERSE 1; INK 3; FLASH 1;"TRA
CCIAMENTO"
4080 LET max=INT (VAL A$(1)): LET min=max
4090 FOR I=2 TO NY
4100 LET n1=INT (VAL A$(I))
4110 IF n1>max THEN LET max=n1
4120 IF n1<min THEN LET min=n1
4130 BEEP .05,0
4140 NEXT I
4150 LET ran=max-min
4160 LET rat=INT (ran/152+.5)
4170 LET ox=40
4180 FOR I=1 TO NY
4190 LET n1=INT (VAL A$(I)): LET nx=INT ((n1-min)/
      rat+.5)+40

```

```

4200 LET C(I)=nx-ox: LET ox=nx
4210 BEEP .05,24
4220 NEXT I
4230 PRINT AT 2,12;"▲▲▲▲▲▲▲▲▲▲"
4500 REM
4510 REM ridisegna la curva di vendita
4520 FOR I=1 TO NY
4530 LET ln=I+3-1: PRINT AT ln,0;Y$(I);AT ln,25;A$(
(I)
4540 NEXT I
4550 PLOT 40,155: DRAW 151,0: PLOT 40,155: DRAW 0,
-155
4560 PLOT 40+C(1),147
4570 FOR I=2 TO NY
4580 DRAW C(I),-8
4590 NEXT I
4600 INPUT "Lo copio su stampante? (s o n)";k$
4610 IF k$="s" OR k$="S" THEN COPY : GO TO 4600
4620 IF k$="n" OR k$="N" THEN RETURN
4630 IF k$="E" THEN STOP
4640 GO TO 4600
7000 REM
7010 REM regolazione numero
7030 LET X$="▲▲▲▲▲▲▲▲▲▲": REM 10 spazi
7040 LET W$="1000000": REM 6 posti decimali
7050 LET T$=STR$(INT VAL S$)
7060 IF ns>0 THEN LET T$=X$(1 TO ns-LEN T$)+T$+"
": GO TO 7080
7070 LET T$=".0"
7080 LET n3=VAL W$(1 TO nd+1): REM nd e' il numero
di posti decimali
7090 LET U$=STR$(INT ((VAL S$-VAL T$)*n3+.5))
7100 IF VAL U$<VAL W$(1 TO nd) THEN LET U$=W$(2
TO nd-LEN U$+1)+U$
7110 IF ns=0 THEN LET T$="."
7120 LET S$=T$+U$
7130 RETURN
8000 REM
8010 REM ciclo principale
8020 LET hm=0: LET cm=hm: LET gm=hm: REM iniz modo
visualizzazione
8030 GO SUB 1000: REM input numero anni
8040 GO SUB 2000: REM input storia di vendita
8050 GO SUB 3000: REM cacola
8060 GO SUB 4000: REM plot grafico
8070 GO TO 8040

```


Valutazione dei beni

Copyright (c) Beam Software

DESCRIZIONE

Questo è uno dei tre programmi di tipo finanziario presentati nel libro. Questo in particolare consente all'utente di dettagliare e valutare i suoi beni personali. La valutazione è utile per assicurazioni che coprono rischi e pericoli di fuoco, o perdite, per furto. Potrebbe anche essere usato per spese per articoli in piccole attività, come uffici o attrezzature di laboratori artigianali.

Ogni bene è valutato sul suo costo originale, valore di rivendita e costo di sostituzione.

Il costo di rivendita è calcolato secondo la retta di svalutazione, e il costo di sostituzione è basato sul locale tasso di inflazione.

Il valore di sostituzione potrebbe anche essere valutato usando un tasso accettato di rivalutazione del prezzo al posto del tasso di inflazione locale.

COME FAR ESEGUIRE IL PROGRAMMA

L'intero programma è, ancora, pilotato dallo schermo. Dovete inserire le INFORMAZIONI DI BASE prima che possiate inserire qualsiasi informazione sugli articoli che possedete.

INFORMAZIONE DI BASE -:

| | |
|-------------------|---|
| NOME | : 20 caratteri |
| ANNO CORRENTE | : 4 caratteri, deve essere per lo meno il 1982 |
| INFLAZIONE LOCALE | : meno del 100% |
| NUMERO DI VOCI | : meno di 100 |

INFORMAZIONI SULLE VOCI -:

| | |
|-----------------|--------------------------------------|
| DESCRIZIONE | : 18 caratteri |
| LOCAZIONE | : 10 caratteri |
| DATE D'ACQUISTO | : 4 caratteri prima di quest'anno |
| VITA | : entro 100 anni |
| COSTO ORIGINALE | : numerico |

STRUTTURA DEL PROGRAMMA

La struttura del programma è la seguente:

VALUTAZIONE DEI BENI

```

100 REM VALUTAZIONE
110 REM Programma di valutazione dei beni
120 GO TO 9000
1000 REM
1010 REM Routine di visualizzazione del titolo
1020 REM
1030 REM Visualizza lo schermo
1040 CLS : PRINT AT 1,6;"VALUTAZIONE DEI BENI":
      PRINT AT 2,6;"-----"
1050 PRINT AT 4,3;"NOME: "
1060 PRINT AT 6,3;"ANNO IN CORSO:"
1070 PRINT AT 8,3;"INFLAZ. ATTUALE:.....%"
1080 PRINT AT 10,3;"NUMERO ARTICOLI: "
1090 PRINT AT 12,1;"TOTALE": PRINT AT 13,1;"-----"
      "
1100 PRINT AT 14,3;"VALORE DI SOSTITUZ.£"
1110 PRINT AT 15,3;"VALORE ORIGINALE...£"
1120 PRINT AT 16,23;"-----"
1130 PRINT AT 17,3;"DIFFERENZA";AT 17,22;"£"
1140 PRINT AT 18,23;"===== "
1150 PRINT AT 19,3;"pc CAMBIO";AT 19,14;".....%"
1160 PRINT AT 21,3;"VALORE TOTALE ATTUALE £"
1170 RETURN
5000 REM
5010 REM Record routine
5020 REM Routine visualizzazione record
5030 FOR I=1 TO NI
5040 CLS : PRINT AT 1,8;"DESCRIZIONE ARTICOLO":
      PRINT AT 2,8;"-----"
5050 PRINT AT 4,1;"DESCRIZIONE:"
5060 PRINT AT 6,1;"LUOGO:"
5070 PRINT AT 8,1;"DATA D'ACQUIS.:";AT 8,22;"VITA:
      ^^"
5080 PRINT AT 10,1;"COSTO ORIG.:£"
5090 PRINT AT 13,1;"ATTUALE": PRINT AT 14,1;"-----"
      "--"
5100 PRINT AT 16,3;"VALORE DI RIVENDITA";AT 16,23;
      "£"
5110 PRINT AT 18,3;"COSTO DI SOSTITUZIONE"
5120 PRINT AT 4,14; INVERSE 1;"....."
5130 PRINT AT 6,11; INVERSE 1;"....."
5140 PRINT AT 8,16; INVERSE 1;"....."
5150 PRINT AT 8,27; INVERSE 1;"^^"
5160 PRINT AT 10,16; INVERSE 1;"....."
5170 REM inserimento dati articoli
5180 PRINT AT 4,0; FLASH 1;">"
5190 INPUT D$(I): IF D$(I,1)="*" THEN GO TO 5410

```

```

5200 PRINT AT 4,14; INVERSE 1;D$(I)
5210 PRINT AT 4,0;" ": PRINT AT 6,0; FLASH 1;">"
5220 INPUT L$(I): IF L$(I)="*" THEN GO TO 5410
5230 PRINT AT 6,11; INVERSE 1;L$(I)
5240 PRINT AT 6,0;" ": PRINT AT 8,0; FLASH 1;">"
5250 INPUT T$(I): IF VAL T$(I,1 TO 4)>VAL C$(1
    TO 4) THEN GO TO 5250
5260 PRINT AT 8,16; INVERSE 1;T$(I)
5270 PRINT AT 8,0;" ": PRINT AT 8,21; FLASH 1;">"
5280 INPUT L: IF L<0 THEN GO TO 5280
5290 LET L=INT L: PRINT AT 8,27; INVERSE 1;L
5300 PRINT AT 8,21;"^": PRINT AT 10,0; FLASH 1;">"
5310 INPUT O: IF O<0 THEN GO TO 5310
5320 LET O=INT O: PRINT AT 10,16; INVERSE 1;O
5330 PRINT AT 10,0;" "
5340 REM calculata il valore di rivendita
5350 LET yu=VAL C$-VAL T$(I): LET yu=(yu<L)*yu+(yu
    >=L)*L: LET rv=INT ((L-yu)*O/L)
5360 LET TC=TC+rv: PRINT AT 16,23+7-LEN (STR$ rv);
    rv
5370 LET rc=INT (O*(1+LI/100)↑yu): LET TR=TR+rc:
    PRINT AT 18,23+7-LEN (STR$ rc);rc
5380 LET TO=TO+O
5390 IF INKEY$="" THEN GO TO 5390
5400 NEXT I
5410 RETURN
6000 REM
6010 REM visualizza titolo finale
6020 PRINT AT 4,9; INVERSE 1;N$
6030 PRINT AT 6,17; INVERSE 1;C$
6040 PRINT AT 8,20; INVERSE 1;INT (LI*100)/100
6050 PRINT AT 10,20; INVERSE 1;NI
6060 PRINT AT 14,24+8-LEN (STR$ TR);TR
6070 PRINT AT 15,24+8-LEN (STR$ TO);TO
6080 LET di=ABS (TR-TO): PRINT AT 17,24+8-LEN (
    STR$ di);di
6090 IF TR<TO THEN PRINT AT 17,21;"-": PRINT AT 1
    9,12;"-"
6100 PRINT AT 19,14;INT ((di/TO)*10000)/100;"^%"
6110 PRINT AT 21,24+8-LEN (STR$ TC);TC
6120 IF INKEY$<>"*" THEN GO TO 6120
6130 RETURN
8000 REM
8010 REM inizializza i dati
8020 DIM C$(4): DIM N$(20): LET N$=" ": LET CY=0:
    LET LI=0: LET NI=0: LET TR=0: LET TO=0:
    LET TC=0
8030 GO SUB 1000: REM visualizzail titolo
8040 PRINT AT 4,9; INVERSE 1;"*****"
8050 PRINT AT 6,17; INVERSE 1;"^"

```

```

8060 PRINT AT 8,20; INVERSE 1;" ";AT 8,23;"^."
8070 PRINT AT 10,20; INVERSE 1;"^."
8080 REM inserimento campi
8090 PRINT AT 4,2; FLASH 1;">"
8100 INPUT N$: IF N$=" " THEN GO TO 8100
8110 PRINT AT 4,9; INVERSE 1;N$(1 TO 20)
8120 PRINT AT 4,2;" ": PRINT AT 6,2; FLASH 1;">"
8130 INPUT C$: IF VAL C$<1982 THEN GO TO 8130
8140 PRINT AT 6,17; INVERSE 1;C$
8150 PRINT AT 6,2;" "
8160 PRINT AT 8,2; FLASH 1;">"
8180 INPUT LI: IF LI>100 OR LI<0 THEN GO TO 8180
8190 PRINT AT 8,20;" "+STR$(INT (LI*100)/100)+
""
8210 PRINT AT 8,2;" ": PRINT AT 10,2; FLASH 1;">"
8220 INPUT NI: PRINT AT 10,20;" "+STR$ NI+"
"
8230 PRINT AT 10,2;"^."
8240 PRINT AT 12,9; FLASH 1; INK 3;"INIZIALIZZAZIO
NE"
8250 DIM D$(NI,18)
8260 DIM L$(NI,10)
8270 DIM T$(NI,4)
8280 DIM L(NI)
8290 DIM O(NI)
8300 FOR I=1 TO NI
8310 LET D$(I)="^": LET L$(I)="^": LET T$(I)="^"
8320 LET L(I)=0: LET O(I)=0
8330 NEXT I
8340 PRINT AT 12,9; INK 3;"FINTO^.....^":
FOR I=1 TO 50: NEXT I
8350 LET TR=0: LET TO=TR: LET TC=TR
8360 RETURN
9000 REM
9010 REM ciclo principale
9020 GO SUB 8000: REM inserisce i dati di testa
9030 GO SUB 5000: REM inserisce articoli
9040 REM visualizza le informazioni finali
9050 GO SUB 1000
9060 GO SUB 6000
9070 PRINT AT 12,9; FLASH 1; INK 1;"IL PROGRAMMA E
' TERMINATO"
9080 STOP

```


Scacchi

Copyright (c) Clifford Ramshaw e Beam Software

Questo programma vi permetterà di sfidare il computer in una partita a scacchi. Ma sappiate che il computer è un giocatore terribile ed è molto molto lento.

Comunque questo programma ha una grafica carina e per chiunque sia un patito di scacchi non dovrebbe essere troppo difficile trasformare il computer in un ottimo giocatore. In alternativa, il programma potrebbe essere modificato per permettere di giocare a due giocatori umani.

A voi la scelta.

Questo programma è un esempio eccellente di cos'è possibile realizzare sullo Spectrum. Gli scacchi sono generalmente considerati un gioco difficile da giocare per i computers. Mentre questo programma non pretende in alcun modo di giocare bene, esso mostra solo cosa può essere fatto col BASIC SPECTRUM con un piccolo sforzo.

Quando il programma viene fatto partire, il computer definirà le variabili usate, disegnerà la scacchiera e i pezzi e poi chiederà una mossa. Voi siete i bianchi (in alto) e la mossa viene inserita come lettera da 'a' ad 'h' seguite da un numero da '1' a '8'.

Il computer non controlla per vedere se la vostra mossa è valida così state attenti, (a meno che siete senza scrupoli e vogliate barare).

Come lavora il programma

La seguente è una descrizione dettagliata del programma

| <u>LINEE</u> | <u>DESCRIZIONE</u> |
|--------------|--|
| 1-4 | inizializzazione dello schermo e di alcune variabili che sono usate per risparmiare spazio nelle istruzioni DATA usate per creare i caratteri grafici dei pezzi degli scacchi. |
| 10-270 | crea i pezzi degli scacchi nel set dei caratteri grafici definibili dall'utente dalle istruzioni DATA alle linee 30-220. |
| 280-315 | disegna la scacchiera vuota sullo schermo. |
| 318-360 | dimensiona il vettore scacchiera e i vettori usati per definire le mosse permesse dei pezzi, poi riempie il vettore scacchiera. |
| 370 | chiama la subroutine per mettere i pezzi sullo schermo. |
| 380-500 | costruisce i vettori usati per definire le mosse permesse dei pezzi. |
| 600-630 | ciclo principale |
| | 1 riceve la mossa del giocatore |
| | riceve la mossa del computer |
| | vai a 1 |

800-900 inserisce la mossa del giocatore.
 1000-1010 disegna il pezzo 'c' alla posizione sulla scacchiera x, y
 3000-3150 inserisce la mossa del giocatore e la visualizza sullo schermo.
 4000-4410 fa la mossa del computer
 5000-5090 controllo per lo scacco matto del computer.

SCACCHI

```

1 PAPER 7: BORDER 7: CLS
2 LET p=248: LET t=31: LET a=128: LET g=13:
  LET h=160
3 LET b=192: LET d=224: LET e=240: LET f=15
4 LET x=0: LET y=1: LET z=3: LET c=7: LET i=160

5 LET f$="": DIM A$(1,1)
6 LET kx=5: LET ky=8: LET q=0
10 FOR m=USR "a" TO USR "t"+7
20 READ n: POKE m,n: NEXT m
30 DATA x,x,x,x,x,x,y,z
40 DATA x,x,x,x,x,x,a,b
50 DATA c,c,z,y,z,c,x,x
60 DATA d,d,b,a,b,d,x,x
70 DATA y,c,c,y,g,f,f,c
80 DATA a,d,d,a,176,e,e,d
90 DATA z,z,z,y,c,f,31,x
100 DATA b,b,b,a,d,e,p,x
110 DATA 4,5,2,y,z,c,c,c
120 DATA 32,h,64,a,b,d,d,d
130 DATA z,y,y,y,z,c,f,x
140 DATA b,a,a,a,b,d,e,x
150 DATA y,z,c,c,14,g,c,z
160 DATA a,b,h,112,e,d,d,b
170 DATA x,x,x,x,x,5,5,c
180 DATA x,x,x,x,x,h,h,d
190 DATA x,x,x,y,z,c,c,z
200 DATA x,x,a,b,d,e,p,152
210 DATA y,y,y,z,c,c,f,x
220 DATA a,a,a,b,d,d,e,x
230 DIM t$(7,2): DIM b$(7,2)
240 FOR x=1 TO 7: READ t$(x),b$(x): NEXT x
250 DATA "AB","CD","QR","ST"
260 DATA "MN","KL","OP","CD"
270 DATA "IJ","KL","EF","GH","^","^"
280 LET i=2: LET p=4: PRINT "^^A.B.C.D.E.F.G.H"
290 PRINT : FOR x=0 TO 7: FOR y=0 TO 1: PRINT ";"
  ^";
300 FOR z=0 TO 3: PRINT PAPER p;"^"; PAPER i;"^
  ^";

```



```

310 NEXT z: PRINT : NEXT y: LET t=i: LET i=p:
    LET p=t: NEXT x
315 FOR y=1 TO 8: PRINT AT y+y,0;y: NEXT y
320 DIM b(8,8): FOR y=1 TO 8: FOR x=1 TO 8
330 LET z=(y=8)*-b(x,1)+(y=2)*-1+(y=7)
340 IF y=1 THEN READ z
350 LET b(x,y)=z: NEXT x: NEXT y
360 DATA -4,-2,-3,-5,-6,-3,-2,-4
370 GO SUB 2000
380 DIM d(6,8,2): DIM p(6)
390 FOR x=1 TO 6: LET p(x)=2: FOR y=1 TO 8
400 LET d(x,y,2)=((y<3) OR (y=8))+((y>3) AND (y<7
    ))*-1
410 LET d(x,y,1)=((y>1) AND (y<5))+((y>5)*-1
420 NEXT y: IF x<>2 THEN GO TO 460
430 FOR y=1 TO 4: READ d(x,y,1),d(x,y,2): LET d(x
    ,y+4,1)=-d(x,y,1): LET d(x,y+4,2)=-d(x,y,2)
440 NEXT y
450 DATA -2,1,-1,2,1,2,2,1
460 IF x=1 THEN LET p(x)=8
470 IF x>4 THEN LET p(x)=1
480 NEXT x
490 DIM c$(5): FOR x=1 TO 5: READ c$(x): NEXT x
500 DATA "p","n","b","r","q"
600 GO SUB 3000
610 PRINT AT 8,19; FLASH 1;"STO' PENSANDO";
620 GO SUB 4000
630 GO TO 600
800 LET A$(1)=INKEY$: IF A$(1)="p" THEN GO TO 90
    0
805 IF (A$(1)="") OR (A$(1)<"a") OR (A$(1)>"h")
    THEN GO TO 800
810 PRINT A$(1);" ";
820 LET X1=CODE A$(1)-96
830 LET A$(1)=INKEY$: IF (A$(1)="") OR (A$(1)<"1"
    ) OR (A$(1)>"8") THEN GO TO 830
840 PRINT A$(1);
850 LET Y1=CODE A$(1)-48
860 RETURN
900 LET X1=999: LET Y1=0: RETURN
1000 LET z1=INT ((ATTR (y+y,x+x))/8): IF z1>=8
    THEN LET z1=z1-8
1010 PRINT BRIGHT 1; PAPER z1; INK i;AT y+y,x+x;t
    $(c);AT y+y+1,x+x;b$(c);
1020 RETURN
2000 FOR y=1 TO 8: FOR x=1 TO 8
2010 LET z=b(x,y): IF z=0 THEN GO TO 2040
2020 LET i=7: IF z>0 THEN LET i=0
2030 LET c=ABS z: GO SUB 1000
2040 NEXT x: NEXT y

```

```

2050 RETURN
3000 IF f$="1" THEN PRINT AT 20,19;"SCACCO MATTO"
      : STOP
3010 PRINT AT 3,20;"La tua mossa": PRINT AT 4,22;"
      DA ";; GO SUB 800: IF X1=999 THEN STOP
3020 LET X=X1: LET Y=Y1
3030 PRINT AT 5,23;"A ";; GO SUB 800: LET XB=X1:
      LET YB=Y1
3040 IF b(XB,YB)>0 THEN LET p(b(XB,YB))=p(b(XB,YB
      ))-1
3050 LET b(XB,YB)=b(X,Y)
3060 LET b(X,Y)=0: LET i=7
3070 IF b(XB,YB)<-1 OR YB<8 THEN GO TO 3120
3080 PRINT AT 6,21;"Pezzo ";
3090 LET A$(1)=INKEY$:
3100 LET a=0: FOR z=1 TO 5: IF c$(z)=A$(1) THEN
      LET a=-z
3110 NEXT z: IF a=0 THEN GO TO 3090
3115 PRINT A$(1);: LET b(XB,YB)=a
3120 LET c=7: GO SUB 1000
3130 LET c=-b(XB,YB): LET x=XB: LET y=YB: GO SUB 1
      000
3140 PRINT AT 3,20;".....";AT 4,22;".....
      .";AT 5,23;".....";AT 6,21;".....";
3150 RETURN
4000 LET xb=0: LET yb=xb: LET db=xb: LET cl=1:
      LET bp=xb: LET x=9
4010 LET ax=kx: LET ay=ky: LET g$="": GO SUB 5010:
      LET g$=f$
4020 FOR y=1 TO 8: FOR x=1 TO 8: LET tb=0: LET cz=
      1: LET ty=0: LET tx=0: LET d=0
4030 IF b(x,y)<1 THEN GO TO 4370
4040 LET p=b(x,y): FOR i=1 TO 8: LET dy=d(p,i,2):
      LET dx=d(p,i,1): LET ax=kx: LET ay=ky: LET c
      l=1: LET po=0
4050 IF x+dx<1 OR x+dx>8 OR y+dy<1 OR y+dy>8
      THEN GO TO 4300
4060 IF p>2 AND p<6 THEN GO TO 4210
4070 IF dy>-1 AND p=1 THEN GO TO 4300
4080 IF y>1 OR p>1 THEN GO TO 4130
4090 LET p(5)=p(5)+1
4100 LET b(x,y)=5
4120 GO TO 4300
4130 IF dx<>0 AND b(x+dx,y+dy)>-1 THEN GO TO 43
      00
4140 IF p=6 THEN LET kx=x: LET ky=y: LET ax=x+dx:
      LET ay=y+dy: GO SUB 5000: IF f$="1" THEN
      GO TO 4300
4150 IF dx=0 AND b(x,y+dy)<>0 THEN GO TO 4300
4160 LET po=8-p-b(x+dx,y+dy)*3

```

```

4170 IF y<>7 OR dx<>0 OR b(x,5)<>0 OR p>1
      THEN GO TO 4300
4180 IF g$="1" THEN GO SUB 5000: IF f$="" THEN
      GO TO 4300
4190 IF g$="1" OR RND>.3 THEN LET c1=2
4200 GO TO 4300
4210 LET x1=x: LET y1=y
4220 IF p=3 AND INT (i/2)*2<i THEN GO TO 4300
4230 IF p=4 AND INT (i/2)*2=i THEN GO TO 4300
4240 IF x1+dx<1 OR x1+dx>8 OR y1+dy<1 OR y1+dy>8
      THEN LET c1=c1-1: GO TO 4300
4250 IF b(x1+dx,y1+dy)>0 THEN LET c1=c1-1:
      GO TO 4300
4260 LET po=8-p+INT (RND*3)-b(x1+dx,y1+dy)*3
4270 IF g$="1" THEN GO SUB 5000: IF f$="" THEN
      LET po=po+50: GO TO 4340
4280 IF b(x1+dx,y1+dy)<>0 THEN GO TO 4300
4290 LET c1=c1+1: LET x1=x1+dx: LET y1=y1+dy:
      GO TO 4240
4300 IF po=0 THEN GO TO 4340
4310 IF g$="1" THEN GO SUB 5000: IF f$="" THEN
      LET po=po+50: GO TO 4340
4320 IF g$="" THEN GO SUB 5000: IF f$="1" THEN
      LET po=0: GO TO 4340
4330 IF po>tb THEN LET ax=x+dx*c1: LET ay=y+dy*c
      1: GO SUB 5000: IF f$="1" THEN LET po=po-p*2
4340 IF po>tb THEN LET tb=po: LET tx=x: LET ty=y:
      LET d=i: LET cz=c1
4350 NEXT i
4360 IF tb>bp OR (tb=bp AND (RND>.9 OR (ty<yb
      AND RND>.5))) THEN LET bp=tb: LET xb=tx:
      LET yb=ty: LET c1=cz: LET db=d
4370 NEXT x: NEXT y: LET p=b(xb,yb): LET x1=xb+d(p
      ,db,1)*c1: LET y1=yb+d(p,db,2)*c1: IF p=6
      THEN LET kx=x1: LET ky=y1
4380 LET ax=kx: LET ay=ky: LET b(x1,y1)=b(xb,yb):
      LET b(xb,yb)=0: LET q=1: GO SUB 5010: LET q=
      0
4385 PRINT AT 8,22;"AAAAAAAAA";AT 9,20;"La mia moss
      a";AT 10,22;"DA ";CHR$(xb+95);"-";CHR$(yb+4
      8);
4386 PRINT AT 11,23;"A ";CHR$(x1+95);"-";CHR$(y1
      +48);
4390 LET i=0: LET c=7: LET x=xb: LET y=yb:
      GO SUB 1000
4400 LET c=b(x1,y1): LET x=x1: LET y=y1: GO SUB 10
      00
4410 RETURN

```

```

5000 LET pc=b(x+dx*c1,y+dy*c1): LET b(x+dx*c1,y+dy
    *c1)=b(x,y): LET b(x,y)=0
5010 LET f$="": FOR r=1 TO 6: FOR j=1 TO 8: LET d1
    =d(r,j,1): LET d2=d(r,j,2): LET l=1
5020 IF ax+d1*l<1 OR ax+d1*l>8 OR ay+d2*l<1 OR ay+
    d2*l>8 THEN GO TO 5060
5030 IF r=1 AND d2<1 AND q=0 THEN GO TO 5060
5040 IF b(ax+d1*l,ay+d2*l)=-r THEN LET f$="1":
    GO TO 5070
5050 IF r>2 AND r<6 AND b(ax+d1*l,ay+d2*l)=0
    THEN LET l=l+1: GO TO 5020
5060 NEXT j: NEXT r
5070 IF x>8 THEN GO TO 5090
5080 LET b(x,y)=b(x+dx*c1,y+dy*c1): LET b(x+dx*c1,
    y+dy*c1)=pc
5090 RETURN

```

Dama

Questo programma combina il BASIC e il linguaggio macchina per giocare abbastanza bene a dama.

Il computer mostra una damiera grafica con i vostri pezzi (rossi) in basso e i suoi pezzi (neri) in alto. Voi muovete sempre per primo e la vostra mossa viene inserita con un "valore da" che dice al computer quale pezzo muovere e un "valore a" che dice al computer dove muoverlo.

I quadrati sono identificati da una lettera e da un numero. Le lettere vanno da "a" ad "h" e sono le colonne della damiera. I numeri vanno dall'1 all'8 e sono le righe. Per esempio una mossa d'apertura potrebbe essere "da a6, a b5" (scriverete 'a6' e 'b5').

Se dovesse capitarvi di perdere la partita o se desiderate ritirarvi dovete battere "0" come primo tasto quando il computer vi chiede la mossa. Di tutte le vostre mosse viene controllata la validità, quindi non potete barare; ma le prese non sono obbligatorie (per voi), e i salti multipli non sono permessi. Il computer prende sempre, se può.

Voi fate damone quando uno dei vostri pezzi raggiunge l'ultima linea. I demoni si muovono solo di un quadrato e possono solo fare delle prese singole ma possono muoversi avanti e indietro (e possono venir mangiati dalle pedine [N.d.T.]).

Come funziona:

La parte in BASIC del programma adempie le funzioni di riordinamento. Definisce la scacchiera, accetta ed esegue le mosse del giocatore e mostra graficamente sullo schermo tutte le mosse. Viene usato un breve programma in linguaggio macchina per le mosse del computer. Ciò è stato fatto per permettere di giocare la partita senza dover aspettare per sempre che il computer faccia la sua mossa. Per coloro che conoscono il linguaggio macchina è accluso un listato di questa parte del programma.

Se non capite il linguaggio macchina, non preoccupatevi poiché questa parte del programma è stata introdotta sotto forma di istruzioni DATA alle linee 9000-9450. Fate solo attenzione quando li introducete.

Come parte della sezione in linguaggio macchina il computer ha una rappresentazione della damiera nella memoria sotto forma di una serie di bytes. Ecco perché vengono usate le PEEK e le POKE per fare cambiamenti anziché usare un vettore. È più facile per il linguaggio macchina accedere a questa damiera in memoria piuttosto che ad un vettore.

IL PROGRAMMA

| <u>LINEE</u> | <u>DESCRIZIONE</u> |
|--------------|---|
| 10-20 | Queste due linee stabiliscono il punto più alto occupato dalla memoria cosicché il programma in linguaggio macchina sarà protetto. Viene quindi chiamata la subroutine a 9000 per introdurre il linguaggio macchina in memoria. |
| 30-79 | Questa sezione stabilisce gli attributi dello schermo e crea la rappresentazione della damiera in memoria |
| 80-160 | Definisce i caratteri grafici definibili dall'utente per i pezzi poi mostra sullo schermo la damiera vuota. |
| 170-230 | Mette i pezzi sullo schermo nella posizione iniziale. |
| 300-320 | Ciclo principale |
| | 1 Accetta la mossa del giocatore |
| | Accetta la mossa del computer |
| | vai a 1 |
| 1000-1060 | Disegna un pezzo sulla damiera nella posizione x, y con colore c. Il pezzo è determinato da p : 0 = spazio; 1 = pedine; 2 = damone. |
| 2000-2540 | Inserimento mossa giocatore |
| | Può essere suddivisa come segue: |
| | 2000-2020 visualizza il messaggio |
| | 2030-2050 accetta il valore "da" e controlla se è valido |
| | 2060-2110 accetta il valore "a" e controlla se è una direzione valida per la mossa |
| | 2120-2200 fa la mossa del giocatore se si tratta di una presa |
| | 2500-2540 fa una mossa normale per il giocatore |
| 3000-3110 | Chiama il programma in linguaggio macchina per la mossa del computer e fa la mossa sulla scacchiera |
| 3500 | Mostra sul video i valori x, y della mossa del computer |
| 3900-3904 | Il programma salta qui se il computer ha perso |
| 4000-5010 | Accetta un valore x, y da usare nella mossa del giocatore |
| | Se il giocatore batte 0 come primo tasto il gioco termina e gli verrà chiesto se vorrà giocare ancora |
| 9000-9020 | Legge il programma in linguaggio macchina dalle istruzioni DATA e lo inserisce nella memoria |
| 9100-9450 | Comprende le istruzioni data che compongono il programma. |

DAMA

```
10 CLEAR 32019
20 GO SUB 9000
30 RESTORE 70
40 BORDER 7: PAPER 7: INK 0: CLS
50 FOR x=32420 TO 32420+89
60 READ a: POKE x,a: NEXT x
70 DATA 255,255,255,255,255,255,255,255,255.
71 DATA 0,1,0,1,0,1,0,1,255
72 DATA 1,0,1,0,1,0,1,0,255
73 DATA 0,1,0,1,0,1,0,1,255
74 DATA 0,0,0,0,0,0,0,0,255
75 DATA 0,0,0,0,0,0,0,0,255
76 DATA 2,0,2,0,2,0,2,0,255
77 DATA 0,2,0,2,0,2,0,2,255
78 DATA 2,0,2,0,2,0,2,0,255
79 DATA 255,255,255,255,255,255,255,255,255
80 FOR x=USR "a" TO 63+USR "a"
90 READ a: POKE x,a: NEXT x
100 DATA 0,7,31,63,63,127,127,127
101 DATA 0,224,248,252,252,254,254,254
102 DATA 127,127,127,63,63,31,7,0
103 DATA 254,254,254,252,252,248,224,0
104 DATA 0,7,31,60,56,116,98,97
105 DATA 0,224,248,60,28,46,70,134
106 DATA 97,98,116,56,60,31,7,0
107 DATA 134,70,46,28,60,248,224,0
110 PRINT "A B C D E F G H": PRINT
115 LET d=3: LET l=5
120 FOR y=1 TO 8: PRINT AT y+y,0;y;" ";
130 FOR z=1 TO 2: FOR x=1 TO 4
140 PRINT PAPER d;" "; PAPER l;" ";: NEXT x:
    PRINT : PRINT " ";: NEXT z
150 LET t=d: LET d=1: LET l=t
160 NEXT y
170 LET p=1
180 FOR x=2 TO 8 STEP 2
190 LET c=0: LET y=1: GO SUB 1000: LET y=3:
    GO SUB 1000
200 LET c=2: LET y=7: GO SUB 1000: NEXT x
210 FOR x=1 TO 7 STEP 2
220 LET c=0: LET y=2: GO SUB 1000
230 LET c=2: LET y=6: GO SUB 1000: LET y=8:
    GO SUB 1000: NEXT x
300 GO SUB 2000
310 GO SUB 3000
320 GO TO 300
1000 PAPER 1: INK c: PRINT AT y+y,x+x;
```

```

1005 IF p=0 THEN PRINT " ";
1010 IF p=1 THEN PRINT "AB";
1020 IF p=2 THEN PRINT "EF";
1030 PRINT AT y+y+1,x+x;
1035 IF p=0 THEN PRINT " ";
1040 IF p=1 THEN PRINT "CD";
1050 IF p=2 THEN PRINT "GH";
1060 RETURN
2000 PAPER 7: INK 0: PRINT AT 6,20;"La tua mossa"
2010 BEEP .5,10: PRINT AT 7,22;"DA.:.:";
2020 PRINT AT 8,22;"A.:.:";
2030 LET l=7: GO SUB 4000: LET fx=x: LET fy=y
2040 LET fp=32419+9*fy+fx: LET f=PEEK fp
2050 IF f<>2 AND f<>130 THEN GO TO 2010
2060 LET l=8: GO SUB 4000: LET tx=x: LET ty=y
2070 LET tp=32419+9*ty+tx: LET t=PEEK tp
2080 LET dx=tx-fx: LET dy=ty-fy
2090 IF (ABS dx<>1 AND ABS dx<>2) OR ABS dx
    <>ABS dy THEN GO TO 2010
2100 IF (dy=1 OR dy=2) AND f=2 THEN GO TO 2010
2110 IF ABS dx=1 THEN GO TO 2500
2120 IF t<>0 THEN GO TO 2010
2130 LET jx=fx+dx/2: LET jy=fy+dy/2
2140 LET jp=32419+9*jy+jx: LET j=PEEK jp
2150 IF j<>1 AND j<>129 THEN GO TO 2010
2160 LET c=2: LET p=0: LET x=fx: LET y=fy:
    GO SUB 1000: POKE fp,0
2170 LET p=1: IF f=130 OR ty=1 THEN LET p=2
2180 LET x=tx: LET y=ty: GO SUB 1000: POKE tp,f:
    IF ty=1 THEN POKE tp,130
2190 LET p=0: LET x=jx: LET y=jy: GO SUB 1000:
    POKE jp,0
2200 RETURN
2500 IF t<>0 THEN GO TO 2010
2510 LET p=0: LET c=2: LET x=fx: LET y=fy:
    GO SUB 1000: POKE fp,0
2520 LET p=1: IF f=130 OR ty=1 THEN LET p=2
2530 LET x=tx: LET y=ty: GO SUB 1000: POKE tp,f:
    IF y=1 THEN POKE tp,130
2540 RETURN
3000 LET mp=USR 32020
3002 PAPER 7: INK 0: PRINT AT 10,20;"La mia mossa"

3003 PRINT AT 11,22;"DA.:.:";
3004 PRINT AT 12,22;" A : ";
3005 FOR i=1 TO 50: NEXT i
3010 IF mp=0 THEN GO TO 3900
3020 LET fy=INT (mp/256): LET ty=mp-256*fy
3030 LET fx=INT (fy/16): LET fy=fy-16*fx
3040 LET tx=INT (ty/16): LET ty=ty-16*tx

```



```

3045 LET x=fx: LET y=fy: LET z=11: GO SUB 3500:
      LET x=tx: LET y=ty: LET z=12: GO SUB 3500
3050 LET p=0: LET x=fx: LET y=fy: GO SUB 1000
3060 LET tp=32419+9*ty+tx: LET t=PEEK tp
3070 LET p=1: IF t=129 THEN LET p=2
3080 LET c=0: LET x=tx: LET y=ty: GO SUB 1000
3090 IF ABS (tx-fx)<>2 THEN RETURN
3100 LET x=fx+(tx-fx)/2: LET y=fy+(ty-fy)/2
3110 LET p=0: GO SUB 1000: RETURN
3500 PRINT AT z,28;CHR$(x+64);CHR$(y+48):
      RETURN
3900 PRINT INK 0;AT 20,0;"Non ho mosse -- HAI VIN
TO"
3901 PRINT AT 21,0;"Un'altra partita ? "
3902 IF INKEY$="" THEN GO TO 3902
3903 IF INKEY$="s" THEN GO TO 50
3904 STOP
4000 PRINT AT 1,28; FLASH 1;" ";CHR$ 8;
4010 LET a$=INKEY$
4015 IF a$="0" THEN GO TO 5000
4020 IF a$<"a" OR a$>"h" THEN GO TO 4010
4030 PRINT a$: LET x=CODE a$-96
4040 PRINT AT 1,29; FLASH 1;"^";CHR$ 8;
4050 LET a$=INKEY$
4060 IF a$<"1" OR a$>"8" THEN GO TO 4050
4070 PRINT a$: LET y=CODE a$-48
4080 RETURN
5000 PRINT INK 0;AT 20,0;"Sei stato sfortunato --
^ HO VINTO"
5010 GO TO 3901
9000 RESTORE 9100: FOR x=32020 TO 32301
9010 READ a: POKE x,a: NEXT x
9020 RETURN
9100 DATA 175,33,177,125,6,10,119,35
9110 DATA 16,252,6,35,33,174,126,126
9120 DATA 254,1,40,16,254,129,32,24
9130 DATA 17,246,255,205,187,125,17,248
9140 DATA 255,205,187,125,17,8,0,205
9150 DATA 187,125,17,10,0,205,187,125
9160 DATA 35,35,16,219,58,182,125,167
9170 DATA 40,18,42,183,125,237,91,185
9180 DATA 125,229,126,54,0,25,54,0
9190 DATA 25,119,24,19,58,177,125,167
9200 DATA 40,29,42,178,125,237,91,180
9210 DATA 125,126,54,0,229,25,119,235
9220 DATA 225,205,135,125,121,230,15,254
9230 DATA 8,192,26,246,128,18,201,1
9240 DATA 0,0,201,235,205,146,125,79
9250 DATA 235,205,146,125,71,201,229,213
9260 DATA 197,17,164,126,175,237,82,61

```

9270 DATA 17,9,0,60,237,82,48,251
9280 DATA 71,25,125,60,7,7,7,7
9290 DATA 128,193,209,225,201,0,0,0
9300 DATA 0,0,0,0,0,0,0,205
9310 DATA 19,126,254,255,200,230,127,254
9320 DATA 1,200,254,2,40,78,229,25
9330 DATA 213,17,246,255,205,13,126,40
9340 DATA 39,17,248,255,205,13,126,40
9350 DATA 31,17,8,0,205,5,126,40
9360 DATA 23,17,10,0,205,5,126,40
9370 DATA 15,209,225,34,178,125,237,83
9380 DATA 180,125,62,1,50,177,125,201
9390 DATA 58,177,125,167,40,235,209,225
9400 DATA 201,205,19,126,230,127,254,2
9410 DATA 201,205,19,126,254,130,201,229
9420 DATA 25,126,225,201,229,25,205,19
9430 DATA 126,225,254,0,192,62,1,50
9440 DATA 182,125,34,183,125,237,83,185
9450 DATA 125,201

LISTATO ASSEMBLY PER LA DAMA

| | | | | | |
|-------------|--------------|------|-------------|--|--|
| | 00110 ; | | | | |
| 7D14 | 00120 | ORG | 32020 | | |
| | 00130 ; | | | | |
| 0000 | 00140 BLANK | EQU | 0 | | ;CODICE PER QUADRATI VUOTI |
| FFFF | 00150 NULL | EQU | -1 | | ;CODICE PER IL BORDO DELLA DAMIERA |
| 0001 | 00160 BLACK | EQU | 1 | | ;CODICE PER I PEZZI NERI |
| 0002 | 00170 WHITE | EQU | 2 | | ;CODICE PER I PEZZI BIANCHI |
| 0080 | 00180 KING | EQU | 128 | | ;SOMMA QUESTO PER UN DAMONE |
| | 00190 ; | | | | |
| 7EA4 | 00200 BOARD | EQU | 32420 | | ;INDIRIZZO DELLA DAMIERA |
| | 00210 ; | | | | |
| 7D14 | 00220 START | EQU | \$ | | |
| 7D14 AF | 00230 | XOR | A | | |
| 7D15 21B17D | 00240 | LD | HL,MOVE | | ;CANCELLA LE VECCHIE MOSSE DALLA MEMORIA |
| 7D18 060A | 00250 | LD | B,10 | | |
| 7D1A 77 | 00260 CLEAR | LD | (HL),A | | |
| 7D1B 23 | 00270 | INC | HL | | |
| 7D1C 10FC | 00280 | DJNZ | CLEAR | | |
| | 00290 ; | | | | |
| 7D1E 0623 | 00300 | LD | B,35 | | ;B=CONTATORE DELLE POSIZIONI DA CONTROLLARE |
| 7D20 21AE7E | 00310 | LD | HL,BOARD+10 | | ;HL = POSIZIONE SULLA DAMIERA |
| | 00320 ; | | | | |
| 7D23 7E | 00330 NEXT | LD | A,(HL) | | ;PRENDE IL PEZZO |
| 7D24 FE01 | 00340 | CP | BLACK | | ;È LA PEDINA DEL COMPUTER |
| 7D26 2810 | 0350 | JR | Z,MANFND | | ;SI = > PEZZO NORMALE |
| 7D28 FE81 | 00360 | CP | BLACK+KING | | ;È IL DAMONE DEL COMPUTER? |
| 7D2A 2018 | 00370 | JR | NZ,ENDSCH | | ;NO => PROVA IL PROSSIMO PEZZO |
| 7D2C 11F6FF | 00380 | LD | DE,-10 | | ;SALTO ALL'INDIETRO |
| 7D2F CDBB7D | 00390 | CALL | CALL TEST | | |
| 7D32 11F8FF | 00400 | LD | DE,-8 | | ;PER UN DAMONE |
| 7D35 CDBB7D | 0410 | CALL | TEST | | |
| 7D38 110800 | 00420 MANFND | LD | DE,8 | | ;TEST SALTA AVANTI PER |
| 7D3B CDBB7D | 00430 | CALL | TEST | | |
| 7D3E 110A00 | 00440 | LD | DE,10 | | ;O UN DAMONE O UNA PEDINA |
| 7D41 CDBB7D | 00450 | CALL | TEST | | |
| 7D44 23 | 00460 ENDSCH | INC | HL | | ;PROSSIMO PEZZO (NECESSARIO CONTROLLARE SOLTANTO |
| 7D45 23 | 00470 | INC | HL | | ;OGNI SECONDA POSIZIONE) |

| | | | | | |
|------|----------|---------|-----------|-------------|---------------------------------------|
| 7D46 | 10DB | 00480 | DJNZ | NEXT | |
| | | 00490 ; | | | |
| 7D48 | 3AB67D | 00500 | LD | A, (CAPT) | CONTROLLA SE È STATA |
| 7D4B | A7 | 00510 | AND | A | ;RILEVATA CATTURA |
| 7D4C | 2812 | 00520 | JR | Z, NORMOV | ;NO=> PROVA UNA MOSSA |
| | | | | | NORMALE |
| 7D4E | 2AB77D | 00530 | LD | HL, (CAPTF) | ;ERA UNA CATTURA PERCIÒ |
| | | | | | FA |
| 7D51 | ED5BB97D | 00540 | LD | DE, (CAPTD) | ;LA MOSSA SULLA DAMIERA |
| 7D55 | E5 | 00550 | PUSH | HL | |
| 7D56 | 7E | 00560 | LD | A, (HL) | ;PEZZO MOSSO |
| 7D57 | 3600 | 00570 | LD | (HL), BLANK | ;ORA UN QUADRATO VUOTO |
| 7D59 | 19 | 00580 | ADD | HL, DE | |
| 7D5A | 3600 | 00590 | LD | (HL), BLANK | ;RIMUOVI IL PEZZO SALTATO |
| 7D5C | 19 | 00600 | ADD | HL, DE | |
| 7D5D | 77 | 00610 | LD | (HL), A | ;METTI INDIETRO IL PEZZO |
| | | | | | NEL NUOVO QUADRATO |
| 7D5E | 1813 | 00620 | JR | GOTMOV | ;RITORNA LA MOSSA AL PRO- |
| | | | | | GRAMMA BASIC |
| 7D60 | | 00630 | NORMOVEQU | \$ | |
| 7D60 | 3AB170 | 00640 | LD | A, (MOVE) | ;CONTROLLO PER LA MOSSA |
| | | | | | TROVATA |
| 7D63 | A7 | 00650 | AND | A | |
| 7D64 | 281D | 00660 | JR | Z, NOMOVE | ;NO => IL COMPUTER PERDE |
| 7D66 | 2AB27D | 00670 | LD | HL, (FROM) | ;ERA UNA MOSSA, COSÌ FAL- |
| | | | | | LA SULLA |
| 7D69 | ED5BB47D | 00680 | LD | DE, (DIR) | ;DAMIERA |
| 7D6D | 7E | 00690 | LD | A, (HL) | ;SALVA IL PEZZO |
| 7D6E | 3600 | 00700 | LD | (HL), BLANK | ;QUADRATO VUOTO |
| 7D70 | E5 | 00710 | PUSH | HL | |
| 7D71 | 19 | 00720 | ADD | HL, DE | |
| 7D72 | 77 | 00730 | LD | (HL), A | ;METTI IL PEZZO NEL NUOVO |
| | | | | | QUADRATO |
| 7D73 | ED | 00740 | GOTMOV EX | DE, HL | ;DE = NUOVA POSIZIONE |
| | | | | | SULLA DAMIERA |
| 7D74 | E1 | 00750 | POP | HL | ;RIPRENDE LA VECCHIA PO- |
| | | | | | SIZIONE SULLA DAMIERA |
| 7D75 | CD877D | 00760 | CALL | CONV | ;CONVERTIRLA PER IL PRO- |
| | | | | | GRAMMA BASIC |
| 7D78 | 79 | 00770 | LD | A, C | |
| 7D79 | E60F | 00780 | AND | 15 | ;CONTROLLO PER PROMO- |
| | | | | | ZIONE |
| 7D7B | FE08 | 00790 | CP | 8 | ;A DAMONE |
| 7D7D | CO | 00800 | RET | NZ | ;NO = > RITORNO AL BASIC |
| TD7E | 1A | 00810 | LD | A, (DE) | ;PRENDE IL PEZZO |
| 7D7F | F680 | 00820 | OR | KING | ;NE FA UN DAMONE |
| 7D81 | 12 | 00830 | LD | (DE), A | |
| 7D82 | C9 | 00840 | RET | | |
| | | 00850; | | | |
| 7D83 | 010000 | 00860 | NOMOVE LD | BC, 0 | ;RITORNA L'INSUCCESSO AL |
| 7D86 | C9 | 00870 | RET | | ;BASIC |
| | | 00880 ; | | | |
| | | 00890 ; | | | |
| | | | | | QUESTA SUBROUTINE PRENDE HL E DE COME |

```

                                POSIZIONE SULLA
00900 ; DAMIERA E LI CONVERTE A COORDINATA X, Y PER IL
                                PROGRAMMA
00910 ; BASIC
00920 ; HL = B (QUESTO SARÀ IL VALORE X,
                                Y DELLA POSIZIONE
00930 ; "DA" PER LA MOSSA DEL
                                COMPUTER)
00940 ;
00950 ; DE = C (X, Y VALORE DELLA POSI-
                                ZIONE "A")
00960 ; IL VALORE X È NELLA METÀ SUPERIORE DEL REGI-
                                STRO
00970 ; (B o C) E IL VALORE Y È NELLA METÀ INFERIORE
00980 ;
7D87 00990 CONV EQU $
7D87 EB 01000 EX DE,HL
7D88 CD927D 01010 CALL CONVIT ;CONVERTE DE
7D8B 4F 01020 LD C,A ;IN C
7D8C EB 01030 EX DE,HL
7D8D CD927D 01040 CALL CONVIT ;CONVERTE HL
7D90 47 01050 LD B,A ;IN B
7D91 C9 01060 RET
                                01070 ;
                                01080 ; CONVERTE HL NELLE COORDINATE X, Y E RITORNA
                                01090 ; IL RISULTATO IN A
                                01100 ; (LA COLONNA DEL PEZZO SULLA DAMIERA)
                                01110 ; È NELLA METÀ SUPERIORE DI A
                                01120 ; IL VALORE Y (LA RIGA DEL PEZZO SULLA DAMIERA)
                                01130 ; È NELLA METÀ BASSA DI A
                                01140 ;
7D92 E5 01150 CONVIT PUSH HL
7D93 D5 01160 PUSH DE ;SALVA REGISTRI
7D94 C5 01170 PUSH BC
7D95 11A47E 01180 LD DE,BOARD ;HL È INDIRIZZO DEL PEZZO
7D98 AF 01190 XOR A ;CUI DEVE ESSERE SOTTRAT-
                                TO
7D99 ED52 01200 SBC HL,DE ;L'INDIRIZZO DELLA DAMIE-
                                RA PER OTTENERE
7D9B 3D 01210 DEC A ;LA POSIZIONE NELLA DA-
                                MIERA
7D9C 110900 01220 LD DE,9 ;LUNGHEZZA DI OGNI LINEA
7D9F 01230 CNVL EQU $
7D9F 3C 01240 INC A ;INCREMENTA NUMERO LI-
                                NEA
7DA0 ED52 01250 SBC HL,DE ;DECREMENTA POSIZIONE
7DA2 30FB 01260 JR NC,CNVL ;SE PIÙ LINEE CONTINUANO
7DA4 47 01270 LD B,A ;B = NUMERO LINEA (Y)
7DA5 19 01280 ADD HL,DE ;RIPRENDI IL NUMERO CO-
                                LONNA
7DA6 7D 01290 LD A,L ;A = NUMERO COLONNA (X)
7DA7 3C 01300 INC A ;LA PRIMA COL È IL NUMERO

```

1

| | | | | | |
|-------------|---------|--|------|--------|--|
| 7DA8 07 | 01310 | RLCA | | | |
| 7DA9 07 | 01320 | RLCA | | | ;MA NELLA METÀ SUPERIORE DEI REGISTRI |
| 7DAA 07 | 01330 | RLCA | | | |
| 7DAB 07 | 01340 | RLCA | | | |
| 7DAC 80 | 01350 | ADD | A,B | | ;SOMMA Y |
| 7DAD C1 | 01360 | POP | BC | | |
| 7DAE D1 | 01370 | POP | DE | | ;RIPRENDE I REGISTRI |
| 7DAF E1 | 01380 | POP | HL | | |
| 7DB0 C9 | 01390 | RET | | | |
| | 01400 ; | | | | |
| | 01410 ; | | | | |
| | 01420 ; | | | | |
| | | AREA DATI PER LE MOSSE NORMALI | | | |
| 7DB1 00 | 01430 | MOVE | DB | 0 | ;FLAG PER INDICARE SE È STATA TROVATA UNA MOSSA |
| 7DB2 0000 | 01440 | FROM | DW | 0 | ;INDIRIZZO SU DAMIERA PER IL PEZZO DA MUOVERE |
| 7DB4 0000 | 01450 | DIR | DW | 0 | ;DIREZIONE DELLA MOSSA |
| | 01460 ; | | | | |
| | 01470 ; | | | | |
| | 01480 ; | | | | |
| | | AREA DATI PER LE CATTURE | | | |
| 7DB6 00 | 01490 | CAPT | DB | 0 | ; FLAG |
| 7DB7 0000 | 01500 | CAPTF | DW | 0 | ;INDIRIZZO DEI PEZZI NELLA DAMIERA |
| 7DB9 0000 | 01510 | CAPTD | DW | 0 | ;DIREZIONE DELLA MOSSA |
| | 01520 ; | | | | |
| | 01530 ; | | | | |
| | 01540 ; | | | | |
| | 01550 ; | | | | |
| | | TEST TIPO DI MOSSE POSSIBILI PER PEZZO ALLA POS. (HL) | | | |
| | 01560 ; | | | | |
| | 01570 ; | | | | |
| | 01580 ; | | | | |
| | | SE UNA MOSSA VALIDA È POSSIBILE VIENE MEMO- RIZZATA | | | |
| | 01590 ; | | | | |
| | 01600 ; | | | | |
| | 01610 ; | | | | |
| | | SE UN PEZZO PUÒ PRENDERE LA SUA MOSSA VIENE MESSA | | | |
| | 01620 ; | | | | |
| | 01630 ; | | | | |
| | 01640 ; | | | | |
| | 01650 ; | | | | |
| | 01660 ; | | | | |
| | | MOSSE VALIDE NORMALI CHE RISULTEREBBERO IN POSSIBILI TRAPPOLE (CATTURA IMMEDIATA) | | | |
| | | SONO MEMORIZZATI SOLO SE NON È STATA TROVATA NESSUNA | | | |
| | 01670 ; | | | | |
| | 01680 ; | | | | |
| | | ALTRA MOSSA | | | |
| 7DBB | 01690 | TEST | EQU | \$ | |
| 7DBB CD137E | 01700 | | CALL | BRDPCE | |
| TDBE FEFF | 01710 | | CP | NELL | ;CONTROLLO PER MOSSA FUORI DALLA DAMIERA |
| 7DC0 C8 | 01720 | | RET | Z | ;NIENTE MOSSA SE È COSÌ |
| 7DC1 E67F | 01730 | | AND | 07FH | |
| 7DC3 FE01 | 01740 | | CP | BLACK | ;OPPURE SU UNA PROPRIA PEDINA |

| | | | | | |
|---------------|---------|------------|------------|--------------------------|---|
| 7DC5 C8 | 01750 | RET | Z | | |
| 7DC6 FE02 | 01760 | CP | WHITE | | CONTROLLO PER LA CATTURA |
| 7DC8 284E | 01770 | JR | Z,EVALCAPT | | ;YOE = CONTROLLO SE VALIDO |
| | 01780 ; | | | | |
| | 11790 ; | | | VALUTA UNA MOSSA NORMALE | |
| | 01800 ; | | | | |
| 7DCA E5 | 01810 | PUSH | HL | | |
| 7DCB 19 | 01820 | ADD | HL,DE | | |
| 7DCC D5 | 01830 | PUSH | DE | | |
| 7DCD 11F6FF | 01840 | LD | DE,-10 | | ; CONTROLLO PER TRAPPOLE |
| 7DD0 CD0D7E | 01850 | CALL | GETPCE | | ; C'È UN DAMONE DIETRO IL PEZZO? |
| 7DD3 2827 | 01860 | JR | Z,TRAP | | ; SI, È UNA TRAPPOLA |
| 7DD5 11F8FF | 01870 | LD | DE,-8 | | |
| 7DD8 CD0D7E | 01880 | CALL | GETOCE | | |
| 7ddb 281F | 01890 | JR | Z,TRAP | | |
| 7DDD 110800 | 01900 | LD | DE,8 | | ;C'È UN PEZZO DAVANTI |
| 7DE0 CD057E | -01910 | CALL | GETPC1 | | ;ALLA PEDINA DEL COMPUTER |
| 7DE3 2817 | 01920 | JR | Z,TRAP | | ;SI => UNA TRAPPOLA |
| 7DE5 110A00 | 01930 | LD | DE,10 | | |
| TDE8 CD057E | 01940 | CALL | GETPC1 | | |
| 7DEB 280F | 01950 | JR | Z,TRAP | | |
| | 01960 ; | | | | |
| 7DED D1 | 01970 | SAVIT POP | DE | | ;NIENTE TRAPPOLE, COSÌ SALVA LA MOSSA |
| 7DEE E1 | 01980 | POP | HL | | |
| 7DEF 22B27D | 01990 | LD | (FROM),HL | | |
| 7DF2 ED53B47D | 02000 | LD | (DIR),DE | | |
| 7DF6 3E01 | 02010 | LD | A,1 | | |
| 7DF8 32B17D | 02020 | LD | (MOVE),A | | ;METTI IL FLAG PER INDICARE UNA MOSSA TROVATA |
| 7DFB C9 | 02030 | RET | | | |
| | 02040 ; | | | | |
| 7DFC | 02050 | TRAP EQU | \$ | | |
| 7DFC 3AB17D | 02060 | LD | A,(MOVE) | | ;ERA UNA TRAPPOLA, COSÌ VEDE SE CI SONO |
| 7DFF A7 | 02070 | AND A | | | ;ALTRE MOSSE |
| 7E00 28EB | 02080 | JR | Z,SAVIT | | ;NO = > SALVARE QUESTA |
| 7E02 D1 | 02090 | POP | DE | | ;ALTRIMENTI IGNORALA |
| 7E03 E1 | 02100 | POP | HL | | |
| 7E04 C9 | 02110 | RET | | | |
| | 02120 ; | | | | |
| 7E05 | 02130 | GETPC1 EQU | & | | ;CONTROLLA SE IL PEZZO A (HL)+(DE) |
| 7E05 CD137E | 02140 | CALL | BRDPCE | | |
| 7E08 E67F | 02150 | AND | 07FH | | È UN PEZZO BIANCO |
| 7E0A FE02 | 02160 | CP | WHITE | | |
| 7E0C C9 | 02170 | RET | | | |
| | 02180 ; | | | | |

| | | | | | | |
|---------------|-------|-------------|------|------------|--|--|
| 7E0D | 02190 | GETPCE | EQU | \$ | | ;CONTROLLA SE IL PEZZO A (HL)+(DE) |
| 7E0D CD137E | 02200 | | CALL | BRDPCE | | |
| 7E10 FE82 | 02210 | | CP | WHITE+KING | | ;È UN DAMONE BIANCO |
| 7E12 C9 | 02220 | | RET | | | |
| | 02230 | | | | | |
| 7E13 E5 | 02240 | BRDPCE | PUSH | HL | | ;RITORNA IL PEZZO AD (HL)+(DE) |
| 7E14 19 | 02250 | | ADD | HL,DE | | |
| 7E15 7E | 02260 | | LD | A,(HL) | | |
| 7E16 E1 | 02270 | | POP | HL | | |
| 7E17 C9 | 02280 | | RET | | | |
| | 02290 | | | | | |
| 7E18 | 02300 | EVALCAPTEQU | | \$ | | ;CONTROLLO PER UNA CAT- TURA VALIDA |
| 7E18 E5 | 02310 | | PUSH | HL | | |
| 7E19 19 | 02320 | | ADD | HL,DE | | |
| 7E1A CD137E | 02330 | | CALL | BRDPCE | | ;PRENDI IL PEZZO SULL'AL- TRA PARTE |
| 7E1D E1 | 02340 | | POP | HL | | ;DEL PEZZO DA CATTURARE |
| 7E1E FE00 | 02350 | | CP | BLANK | | ;È VUOTO? |
| 7E20 C0 | 02360 | | RET | NZ | | ;NO = > NIENTE CATTURE |
| 7E21 3E01 | 02370 | | LD | A,1 | | ;SI = > CATTURA SALVA |
| 7E23 32B67D | 02380 | | LD | (CAPT),A | | |
| 7E26 22B77D | 02390 | | LD | (CAPTF),HL | | |
| 7E29 ED53B97D | 02400 | | LD | (CAPTD),DE | | |
| 7E2D C9 | 02410 | | RET | | | |
| | 02420 | | | | | |

L'avventura

Un'avventura è un programma che vi consente di entrare ed esplorare nuovi strani mondi, trovare tesori e combattere contro mostri feroci. Come avventuriero dovrete risolvere gli enigmi e i problemi del mondo in cui entrate se volete uscirne vivi.

Questo programma vi permette di avere un'avventura sul vostro SPECTRUM. Il computer descriverà i luoghi che visiterete e le cose che vedrete. Inoltre vi inciterà a dire ciò che volete fare. I comandi che potete dare consistono in 1 o 2 parole. Dovete sempre fornire un verbo (cioè l'azione che desiderate fare) e spesso avrete bisogno di un sostantivo (cioè l'oggetto sul quale volete svolgere l'azione).

Alcuni esempi sono:

- 'nord': si dirige a nord e guarda in giro
- 'prendi ascia': se v'è un'ascia dove vi trovate questa dirà al computer di tentare di prenderla per voi.

In alcuni casi vi saranno chieste più informazioni dopo avere dato un'ordine al computer.

Per esempio se dite:

'Uccidi drago' il computer vi chiederà con cosa volete ucciderlo. In questo caso è sufficiente inserire il nome dell'arma che desiderate usare oppure niente (battete solo ENTER) se volete usare le mani nude.

Oltre al fatto di procurarvi un'avventura, questo programma vi permette anche di creare facilmente i vostri mondi, per voi o per un amico, nei quali avventurarvi.

COME GIOCARE

Il computer vi mostrerà sul video una descrizione del luogo in cui vi trovate e descriverà tutti gli oggetti che vedete.

Vi chiederà poi cosa volete fare. Dovete allora rispondere con un ordine del tipo descritto sopra. Il computer cercherà di eseguire il vostro ordine e se tutto va bene, tornerà indietro e farà ciò in continuazione. Se peraltro il vostro comando non funziona, forse a causa di un errore di battitura e se non siete ancora in grado di dare tale ordine, il computer stamperà un messaggio adatto al caso, tornate indietro e riprovate.

Alcuni ordini che potete usare sono:

nord, sud, est, ovest, alto e basso

Questi vi permettono di muovervi intorno.

inventario — fa una lista di tutto quello che trasportate.

prendi — raccoglie gli oggetti

lascia — lascia gli oggetti ecc.

Se nella descrizione dell'oggetto vi è un aggettivo che lo precede, per es. "malvagio folletto", allora non potete dire "uccidi folletto". Dovete usare "uccidi malvagio folletto" o solo "uccidi malvagio".

È da notare che molte parole possono essere abbreviate ad una o due lettere; purché ciò non causi ambiguità fra le parole. Se un'abbreviazione comprende più di una parola il computer usa la prima che incontra.

Quando si gioca un'avventura è di solito una buona idea disegnare una mappa del mondo che si sta esplorando. Poiché molte avventure contengono dei labirinti ed è quasi impossibile trovare la strada a meno che non abbiate una buona mappa che vi indichi le strade da seguire.

COME FUNZIONA IL PROGRAMMA

Questo programma consiste in due parti distinte. In primo luogo vi è il data base che in sostanza è il gioco. Poi vi è il programma necessario per consentire al giocatore di interagire con il data base. Ciò consiste nell'inizializzazione del gioco, nell'accettazione delle mosse dell'utente e nell'esecuzione dell'azione desiderata.

IL DATA BASE

Le parti fondamentali di questa avventura sono i luoghi che compongono il mondo, una mappa che descrive come questi luoghi sono collegati fra di loro, gli oggetti presenti in questo mondo e tutte le azioni che il giocatore può fare. Il mondo in cui ciascuna di queste parti è costituita verrà ora descritto più dettagliatamente.

I LUOGHI

Per ogni luogo abbiamo bisogno di una descrizione da dare al giocatore per dirgli dove si trova e una lista dei luoghi in cui può andare da dove si trova. È spesso altrettanto utile avere qualche tipo di condizione che possa essere applicata entrando in un luogo. Queste possono essere di qualsiasi tipo, accertarsi che il giocatore abbia una fonte di luce prima di entrare, eseguire qualche azione quando entra (come ad esempio farlo attaccare da un mostro se non porta un particolare oggetto).

In questo programma ogni luogo ha un unico numero che il computer usa per accedere a quel luogo. I numeri iniziano da 1 ed aumentano di 1 per ogni luogo fino all'ultimo luogo. Ad esempio questa avventura ha 30 luoghi.

I luoghi esistono nel programma contenuti in istruzioni DATA nella forma
numero di linea DATA "stringa", N, S, E, O, A, B, condizione

Il numero di linea è il numero del luogo più una costante (9000 in questo gioco).

La "stringa" è la descrizione del luogo.

I valori N, S, E, O, A, B sono i luoghi nei quali andrà a finire il giocatore se tenterà di andare nella direzione appropriata (nord, sud, ..., alto, basso). Il valore 0 sta a significare che non vi è alcun luogo verso il quale dirigersi in quella direzione.

La condizione è il numero della condizione da eseguire all'entrata in QUESTO luogo.

Le condizioni consistono in brevi pezzi di programma localizzati al numero della condizione più una costante (7100 in questo programma).

OGGETTI

Le informazioni di cui abbiamo bisogno per ogni oggetto consistono dalla sua descrizione, dove si trova, che tipo di oggetto è (cioè un tesoro o un mostro ecc.), quant'è forte e, per permettere maggior flessibilità, una condizione da eseguirsi se il giocatore prova a prendere questo oggetto. Queste condizioni possono essere una che fallisce sempre nel fermare il giocatore quando tenta di prendere cose come tavoli, oppure una per uccidere il giocatore se prova a prendere un oggetto proibito.

Il computer sa tutto di ciascun oggetto dal suo numero, proprio come per i luoghi.

Il formato dei DATA per gli oggetti è leggermente diverso da quello per le posizioni. Ogni oggetto ha due linee nel programma che lo descrivono.

La prima è

numero di linea DATA «stringa»

in cui

il numero di linea=numero oggetto+una costante (8000 in questa avventura)
e "stringa" è la descrizione dell'oggetto

La seconda è:

numero di linea DATA tipo, forza, condizione

dove

numero di linea=numero oggetto+una costante (8500)

il tipo è 0 per un oggetto comune

>0 per un tesoro e questo valore è il punteggio per questo tesoro

<0 per un mostro

forza

questa è usata in due modi:

per un'arma essa è il valore sommato

alla forza del giocatore se la usa

in una lotta

e per un mostro essa è la sua forza,

che è confrontata con quella del giocatore

in battaglia

condizione è la condizione da eseguirsi se
il giocatore prova a prendere questo oggetto

Per finire, abbiamo bisogno del luogo di ogni oggetto e questo è memorizzato in un vettore. I luoghi iniziali per ogni oggetto (come pure il numero degli oggetti) sono memorizzati in un'istruzione DATA (alla linea 8000 di questo programma) e il vettore luogo viene creato da ciò.

I valori usati per questi luoghi sono:

- 1 — è usato per oggetti che non potete vedere o usare, come animali morti o tesori che sono stati riportati ai luoghi originari
- 0 — è per gli oggetti trasportati dal giocatore
- >0 — tutti gli altri oggetti, e questo valore è il numero del luogo dell'oggetto.

La lista dei verbi che il giocatore può usare è collocata nelle istruzioni DATA (che iniziano a 7001 in questo programma). Ogni volta che un giocatore inserisce un verbo, questa lista viene controllata per trovare il verbo corrispondente.

Ogni verbo ha anche un numero che il programma usa per trovare la subroutine che esegue l'azione del verbo. Il numero del verbo è uguale alla sua posizione nella lista (cioè il primo verbo è il numero 1) e la subroutine per eseguire il verbo si trova a una costante + 20 * il numero del verbo (la costante in questo programma è 1480, così la prima subroutine è a 1500 e le altre ogni 20 linee dopo questa).

La seguente è una descrizione dettagliata del programma.

| <u>LINEE</u> | <u>FUNZIONE</u> |
|--------------|--|
| 5-40 | Inizializzazione le variabili a valore fisso, i parametri dello schermo e le variabili peculiari di questa avventura particolare, vengono definite qui. |
| 50-110 | ciclo di controllo per il programma, consiste in: 1 stampa la descrizione del luogo stampa ogni oggetto visibile 2 accetta l'istruzione dell'utente esegue l'azione che vuole il giocatore se l'azione è fallita vai a 2, vai a 1 |
| 900-940 | inizializzazione delle particolari variabili di questa avventura, cioè definisce il numero di azioni e gli oggetti, costruisce il vettore del luogo dell'oggetto e definisce alcune stringhe che sono comuni a un certo numero di luoghi per risparmiare memoria. Definisce anche alcune variabili che vengono usate come condizioni per il gioco, ad esempio se una porta è chiusa a chiave o meno, ecc. |

- 1000-1100 inserisce l'istruzione del giocatore, le lettere maiuscole vengono convertite a minuscole e le prime due parole che il giocatore inserisce vengono messe nelle variabili a\$ e n\$ (una parola è una sequenza di caratteri che termina con uno spazio).
Si cerca poi nella lista dei verbi a\$ e, se viene trovata, questa subroutine ritorna il numero del verbo al programma principale, altrimenti il giocatore è invitato a provare ancora.
- 1200-1260 questa subroutine cerca in una data lista una data parola. Viene usata per ricercare i verbi per un'azione e le descrizioni degli oggetti per un oggetto.
- 1300-1311 ricerca nelle descrizioni degli oggetti la variabile n\$ (la seconda parola inserita dal giocatore).
Se viene trovata, essa ritorna la descrizione trovata nella lista, il numero dell'oggetto ed un flag per indicare che la ricerca ha avuto successo, altrimenti il flag indica il fallimento.
- 1500-1802 questo set di subroutines viene usato per eseguire le azioni richieste da ogni verbo.
- 1987 si salta a questa linea se alcune azioni falliscono.
Scriva il messaggio "non puoi" e definisce il flag per indicare fallimento.
- 1988 si salta a questa linea se un'azione fallisce e non deve essere stampato il messaggio di cui sopra, si limita a definire il flag.
- 1999 si salta a questa linea se un'azione ha successo, essa definisce il flag per indicare il successo.
- 2000-2050 lista tutti quegli oggetti il cui luogo è uguale alla variabile "l". Se non ne viene trovato alcuno scrive "niente".
- 7000-7004 questa area contiene la lista dei verbi, la linea 7000 contiene il numero dei verbi.
- 7100-7160 queste sono le condizioni che sono poste per prendere oggetti o entrare in luoghi. Notate che la condizione 0 ha sempre successo, così se un oggetto o un luogo non ha bisogno di condizione gli viene data questa.
- 8000-8525 questi sono i DATA per gli oggetti.
- 9001-9030 questi sono i DATA per i luoghi.
- 9980-9981 si salta a queste linee alla fine del gioco se il giocatore ha trovato con successo tutti i tesori.
- 9990-9992 viene usata questa sezione alla fine del gioco (o dopo che il giocatore è stato ucciso o se ha finito il gioco), essa scrive il punteggio e chiede se il gioco deve iniziare nuovamente.

COME CREARSI UNA PROPRIA AVVENTURA

In primo luogo avete bisogno di un mondo da esplorare. Questo implica la creazione nel DATA base delle descrizioni di tutti i luoghi da visitare con i collegamenti fra loro. Insieme a ciò dovrete decidere se e quali condizioni dovranno essere applicate per entrare nei luoghi.

Facciamo un esempio, costruiamoci un mondo con due luoghi.

Questi saranno:

Un corridoio che si estende a perdita d'occhio
Una piccola caverna

Il corridoio è a nord della caverna e a sud il corridoio termina nella caverna. Non sono permesse altre direzioni. Come condizione ci sarà una porta tra i due luoghi che sarà chiusa a chiave. Il giocatore può passare da un luogo all'altro solo se la porta è stata aperta. Così abbiamo:

9001 DATA "in un lungo corridoio che si estende a perdita d'occhio puoi vedere" 0,2,0,0,0,0
9002 DATA "in una piccola caverna" ,1,0,0,0,0,1

Notate che la porta chiude solo la caverna così che entrando nel corridoio non si hanno condizioni mentre per entrare nella caverna è necessario che la porta sia aperta.

Siccome il corridoio è il primo luogo, questo sarà il luogo dal quale il giocatore parte e nel quale andrà messo il tesoro.

Abbiamo ora bisogno di alcuni oggetti. Dobbiamo già avere una porta chiusa e una porta aperta (all'inizio tra i due luoghi ci sarà la porta chiusa, ma quando la porta viene aperta metteremo la porta chiusa nel luogo -1 per sbarazzarci di essa e la sostituiremo con quella aperta).

Ovviamente abbiamo bisogno anche di una chiave per aprire la porta. Per quanto riguarda il tesoro abbiamo uno smeraldo nella caverna. Ci sarà anche una spada nel corridoio e un serpente a guardia dello smeraldo.

Così abbiamo:

| Oggetto | Tipo | Forza | Condizione | Luogo |
|--------------|------|-------|------------|-------|
| Porta chiusa | 0 | 0 | Fallimento | 1 |
| Porta aperta | 0 | 0 | Fallimento | -1 |
| Chiave | 0 | 0 | 0 | 1 |
| Spada | 0 | 10 | 0 | 1 |
| Serpente | -1 | 100 | Fallimento | 2 |
| Smeraldo | 30 | 0 | 2 | 2 |

Questo significa che il giocatore non può prendere né la porta né il serpente e lo smeraldo può essere preso solo se viene soddisfatta la condizione 2. La condizione sarà un test per vedere se il serpente è vivo.

La spada ha una forza di 10 e il serpente di 100 così se il giocatore incomincia con una forza di 100 il combattimento tra loro è abbastanza equilibrato, con un piccolo vantaggio per il giocatore.

Lo smeraldo ha un punteggio di 30, il serpente è un mostro e tutti gli altri sono soltanto oggetti.

Nel programma ciò sarebbe:

```
8000 DATA 6,1,-1,1,1,2,2
8001 DATA "Porta chiusa"
8002 DATA "Porta aperta"
8003 DATA "Chiave"
8004 DATA "Spada"
8005 DATA "Serpente"
8006 DATA "Smeraldo"
8501 DATA 0,0,fa-cs
8502 DATA 0,0,fa-cs
8503 DATA 0,0,0
8504 DATA 0,10,0
8505 DATA -1,100,fa-cs
8506 DATA 30,0,2
```

(fa è l'indirizzo della routine fallimento
e cs è l'indirizzo delle condizioni, così
fa-cs è una condizione che fallisce)

Ora abbiamo bisogno delle azioni da eseguire. I movimenti e le loro subroutines come, "elenca", "prendi", "lascia", "uccidi" sono abbastanza comuni e saranno probabilmente usati in tutte le vostre avventure così, per queste, guardate il programma. L'altra azione che avremo sarà "apri" per la porta, così la lista dei verbi sarà:

```
7000 11 (numero delle azioni)
7001 DATA "nord", "sud", "est", "ovest"
7002 DATA "alto", "basso", "prendi", "lascia"
7003 DATA "uccidi", "elenca", "apri"
```

La sola routine che dobbiamo scrivere è per "apri". Questo deve controllare se il giocatore è nel corridoio, se sta portando la chiave e se la porta è ancora chiusa. Ora abbiamo bisogno di una variabile per indicare se la porta è chiusa o meno, diciamo "lu" (ricordate che questa deve essere inizializzata in principio o la porta può essere aperta mentre noi pensiamo che sia chiusa).

L'apertura, quindi, consisterà in:

```
1700 IF lu=1 OR 1 <> 1 OR o(3) <> 0
      THEN GO TO fa      (controllo condizioni)
1701 LET o(1)=-1        (si disfa della porta chiusa)
1702 LET o(2)=1 (la sostituisce con l'altra)
1703 LET lu=1          (definisce la variabile)
1704 GO TO tr          (successo)
```

Ora abbiamo bisogno delle condizioni, ne abbiamo solo due e sono:

1 è la porta aperta

e

2 è il serpente morto.

Nel programma queste saranno:

```
7101 GO TO fa+lu  (se lu=0 questa andrà a fa per il fallimento;
                  se lu=1 questa andrà a tr per il successo)
7102 GO TO fa+(o(5)=-1)
```

Questa seconda può richiedere alcune spiegazioni.

Il suo significato è

```
IF o(5)=-1 THEN GO TO tr
```

altrimenti

```
IF o(5)<>-1 THEN GO TO fa
```

Questo funziona perché la condizione $o(5)=-1$ sarà uguale a 1 se $o(5)=-1$ e sarà uguale a 0 se $o(5)<>-1$.

Così ora abbiamo un'avventura anche se piuttosto piccola.

Buona fortuna nello scrivere le vostre.

L'AVVENTURA

```
5 LET tr=1999: LET fa=1998: LET ls=9000: LET os
  =8000: LET oz=8500: LET vs=7000: LET cs=7100:
  LET sc=0
6 LET vr=1480: LET z=0: LET ct=1997: LET dr=160
  l: LET cx=6: LET st=100
10 BORDER 5: PAPER 7: INK 0: CLS
30 LET l=1
40 GO SUB 900
50 POKE 23692,255: RESTORE l+ls: READ m$:
  PRINT "Sei ";m$: LET q$="Vedi": GO SUB 2000
60 BEEP .2,12: PRINT
70 GO SUB 1000
80 PRINT v$;" ";
90 GO SUB vr+vn*20
100 IF f=0 THEN GO TO 60
110 PRINT : GO TO 50
900 RESTORE os: READ oc: DIM o(oc)
910 FOR i=1 TO oc: READ o(i): NEXT i
920 RESTORE vs: READ vc
930 LET be=0: LET lu=0: LET gu=0: LET e$="in un d
  eserto senza fine": LET p$="in un passaggio"
940 RETURN
1000 INPUT "Dimmi cosa devo fare ";s$: IF LEN s$=0
  THEN GO TO 1000
1010 LET a$="": LET n$=""
1020 LET x=0: FOR i=1 TO LEN s$
1030 LET i$=s$(i TO i)
1040 IF i$>="A" AND i$<="Z" THEN LET i$=CHR$(32+
  CODE i$)
1050 IF i$=" " AND x=1 THEN LET i=LEN s$: GO TO 1
  080
1051 IF i$=" " THEN LET x=1: GO TO 1080
1060 IF x=0 THEN LET a$=a$+i$
1070 IF x=1 THEN LET n$=n$+i$
1080 NEXT i: LET w$=a$: LET c=vc: LET d=vs+l:
  GO SUB 1200
1090 IF vn=0 THEN PRINT "Non capisco ";a$:
  GO TO 1000
1100 LET v$=y$: RETURN
1200 LET vn=0
1210 RESTORE d
1220 FOR i=1 TO c: READ m$: LET x=LEN m$: IF x>
  LEN w$ THEN LET x=LEN w$
1240 IF w$( TO x)=m$( TO x) THEN LET vn=i: LET y$
  =m$: LET i=c
1260 NEXT i: RETURN
1300 IF n$="" THEN GO TO 1311
```

```

1301 LET w$=n$: LET c=oc: LET d=os+1: GO SUB 1200:
      IF vn=0 THEN GO TO 1311
1310 PRINT y$: GO TO tr
1311 PRINT "Non capisco ";n$: GO TO fa
1500 LET d=1: GO TO dr
1520 LET d=2: GO TO dr
1540 LET d=3: GO TO dr
1560 LET d=4: GO TO dr
1580 LET d=5: GO TO dr
1600 LET d=6
1601 PRINT : RESTORE 1+1s: READ m$: FOR i=1 TO d:
      READ n1: NEXT i
1602 IF n1=0 THEN PRINT "Non puoi andare in quell
a direz.": GO TO fa
1603 RESTORE n1+1s: READ m$,s,s,s,s,s,s
1604 READ s
1605 GO SUB cs+s: IF f=0 THEN GO TO 1611
1610 IF o(1)=0 THEN LET l=n1: GO TO tr
1611 PRINT "Qualcosa ti ferma": GO TO fa
1620 GO SUB 1300: IF f=0 THEN RETURN
1621 IF o(vn)<>1 THEN PRINT "Non e' qui":
      GO TO fa
1622 RESTORE oz+vn: READ s: IF s<0 THEN GO TO ct
1624 READ s
1625 READ s
1626 GO SUB cs+s: IF f=0 THEN GO TO ct
1630 IF cx<>0 THEN LET cx=cx-1: LET o(vn)=0:
      RETURN
1631 PRINT "Stai trasportando troppe cose":
      GO TO fa
1640 GO SUB 1300: IF f=0 THEN RETURN
1641 IF o(vn)<>0 THEN PRINT "Non ce l'hai":
      GO TO fa
1642 RESTORE oz+vn: READ s
1643 LET o(vn)=1: IF l=1 AND s<>0 THEN LET sc=s
c+s: LET o(vn)=-1: PRINT "Il ";y$;" e' svanit
o"
1644 LET cx=cx+1: IF sc=220 THEN GO TO 9980
1645 RETURN
1660 GO SUB 1300: IF f=0 THEN RETURN
1661 LET t$=y$: LET mn=vn: RESTORE oz+vn: READ s,m
s: IF s>=0 THEN GO TO ct
1662 LET s=0: BEEP .1,20: INPUT "uccidere con cosa
?";w$: IF w$="" THEN GO TO 1670
1663 LET c=oc: LET d=os+1: GO SUB 1200: IF vn=0
THEN GO TO 1662
1664 IF o(vn)<>0 THEN GO TO 1662
1665 RESTORE oz+vn: READ s,s: IF s=0 THEN GO TO 1
662
1666 PRINT " con ";y$;

```

```

1670 PRINT : LET s=s+st: IF ms>s+RND*20 THEN
    GO TO 1679
1671 IF s>ms+RND*15 THEN GO TO 1678
1672 PRINT "Il ";t$;" ti attacca di nuovo. Ti sent
    i piu' debole"
1673 LET st=st-RND*5: GO TO tr
1678 PRINT "Hai ucciso il ";t$: PRINT "Il corpo sv
    anisce in una nuvola di fumo": LET o(mn)=-1:
    GO TO tr
1679 PRINT "Il ";t$;" ti ha ucciso": GO TO 9990
1680 LET l1=1: LET l=0: LET q$="Stai trasportando"
    : GO SUB 2000: LET l=l1: GO TO tr
1700 PRINT sc: GO TO tr
1720 GO SUB 1300: IF f=0 THEN RETURN
1721 IF vn<>20 OR o(2)<>0 THEN GO TO ct
1722 LET o(14)=o(vn): LET o(vn)=-1: PRINT "Una chi
    ave d'avorio cade dal soffitto": RETURN
1740 IF l<>l1 AND l<>29 THEN GO TO ct
1741 IF l=l1 AND o(3)<>0 THEN PRINT "Affondate
    nel lago e annegate": GO TO 9990
1742 IF l=l1 THEN LET l=29: GO TO tr
1743 LET l=l1: GO TO tr
1760 GO SUB 1300: IF f=0 THEN RETURN
1761 IF l=l AND o(4)=0 THEN LET o(22)=-1: LET o(2
    3)=1: LET lu=l: RETURN
1762 IF l=13 AND o(14)=0 THEN LET o(24)=-1: LET o
    (25)=1: LET gu=l: RETURN
1763 GO TO ct
1780 IF o(16)<>0 OR l<3 OR l>10 THEN GO TO ct
1781 PRINT "Trovate ";
1782 IF l=6 AND o(4)=-1 THEN LET o(4)=1: PRINT "q
    ualcosa": GO TO tr
1783 PRINT "niente": GO TO tr
1800 GO SUB 1300: IF f=0 THEN RETURN
1801 RESTORE oz+vn: READ s,s1,s1
1802 IF s<0 THEN GO TO ct
1803 GO SUB cs+s1: IF f=0 THEN GO TO ct
1804 PRINT "Questo colpisce la macchia": LET o(vn)
    =-1: RETURN
1820 GO SUB 1300: IF f=0 THEN RETURN
1821 IF l<>24 OR vn<>7 THEN GO TO 1830
1822 PRINT "Appare un ponte sul baratro": LET o(15
    )=1: LET be=l: GO TO tr
1830 PRINT "Non succede niente": GO TO fa
1840 IF l=24 OR l=26 THEN PRINT "La caduta ti ha
    rotto l'osso del collo": GO TO 9990
1841 GO TO 1830
1860 GO SUB 1300: IF f=0 THEN RETURN
1861 IF vn<>18 THEN GO TO ct

```

```

1862 PRINT "Era delizioso": LET o(19)=o(vn): LET o
      (vn)=-1: RETURN
1997 PRINT "Non puoi"
1998 LET f=0: RETURN
1999 LET f=1: RETURN
2000 LET x=0: PRINT q$
2010 RESTORE os+1: FOR i=1 TO oc: READ m$
2020 IF 1<>o(i) THEN GO TO 2030
2021 LET x=1: PRINT " un";
2022 LET z$=m$(1 TO 1): IF z$="s" AND (m$(2)="c"
      OR m$(2)="b") OR z$="z" THEN PRINT "o";
2023 FOR q=1 TO LEN m$: IF m$(q)<>" " THEN
      NEXT q
2024 IF m$(q-1)="a" THEN PRINT "a";
2028 PRINT " ";m$
2030 NEXT i
2040 IF x=0 THEN PRINT " niente"
2050 RETURN
7000 DATA 19
7001 DATA "nord","sud","est","ovest","alto","basso
      "
7002 DATA "prendi","lascia","uccidi","inventario",
      "punteggio"
7003 DATA "taglia","nuota","apri","scava"
7004 DATA "mangia","brandisci","salta","bevi"
7100 GO TO tr
7101 GO TO fa+lu
7102 GO TO fa+gu
7103 GO TO fa+be
7104 GO TO fa+(o(3)=0)
7105 GO TO fa+(o(11)=-1)
7106 GO TO fa+(o(12)=-1)
7107 GO TO fa+(o(10)=-1)
7150 IF o(18)<>0 THEN GO TO tr
7151 LET o(18)=-1: LET o(12)=-1: PRINT "Nonappena
      entrato, un pirata ti ruba il rum e scoppia a
      ridere": GO TO tr
7160 PRINT "Il diavolo ti uccide": GO TO 9990
8000 DATA 25,1,12,27,-1,14,17,21,25,29,12,14,17,20
      ,-1,-1,11,2,2,-1,19,11,1,-1,13,-1
8001 DATA "lampada di ottone"
8002 DATA "spada"
8003 DATA "snorkel"
8004 DATA "grande chiave"
8005 DATA "tappeto persiano"
8006 DATA "moneta d'oro"
8007 DATA "bacchetta d'argento"
8008 DATA "rubino"
8009 DATA "diamante"
8010 DATA "gnomo spregevole"

```

8011 DATA "dragone verde"
8012 DATA "pirata"
8013 DATA "diavolo"
8014 DATA "chiave d'avorio"
8015 DATA "ponte di cristallo"
8016 DATA "doccia"
8017 DATA "tavolo da cucina"
8018 DATA "bottiglia di rum"
8019 DATA "bottiglia vuota"
8020 DATA "fune tesa fra il pavimento e il soffitt
o"
8021 DATA "piccolo lago"
8022 DATA "porta chiusa"
8023 DATA "porta"
8024 DATA "cancello chiuso col lucchetto"
8025 DATA "cancello"
8501 DATA z,z,z
8502 DATA z,20,7
8503 DATA z,z,z
8504 DATA z,z,z
8505 DATA 10,z,5
8506 DATA 50,z,6
8507 DATA 20,z,z
8508 DATA 30,z,z
8509 DATA 100,z,z
8510 DATA -1,87,z
8511 DATA -1,110,z
8512 DATA -1,200,z
8513 DATA -1,200,z
8514 DATA 10,z,z
8515 DATA z,z,z
8516 DATA z,z,z
8517 DATA z,z,fa-cs
8518 DATA z,z,z
8519 DATA z,z,z
8520 DATA z,z,fa-cs
8521 DATA z,z,fa-cs
8522 DATA z,z,fa-cs
8523 DATA z,z,fa-cs
8524 DATA z,z,fa-cs
8525 DATA z,z,fa-cs
9001 DATA "nel soggiorno di una grande casa. Un'in
segna dice porta tutti i tesori qui.",2,z,z,z
,z,12,z
9002 DATA "in cucina",z,1,3,z,z,z,z,z
9003 DATA e\$,z,4,5,2,z,z,z
9004 DATA e\$,3,7,6,z,z,z,z
9005 DATA e\$,z,6,7,3,z,z,z
9006 DATA e\$,5,8,z,4,z,z,z
9007 DATA e\$,z,z,z,4,z,z,z

```

9008 DATA e$,6,9,z,z,z,z,z
9009 DATA e$,8,z,10,z,z,z,z
9010 DATA e$,11,z,z,9,z,z,z
9011 DATA "ad un'oasi",z,10,z,z,z,z,z
9012 DATA "in una cantina",z,16,13,z,1,z,1
9013 DATA "al CANCELLO DELL'INFERNO",z,z,z,12,
z,20,z
9014 DATA "in una caverna oscurata",z,15,z,z,z
,z,z
9015 DATA p$,14,18,z,16,z,z,z
9016 DATA p$,12,z,15,z,z,17,z
9017 DATA "nel covo dei pirati",z,z,z,z,16,z,50
9018 DATA p$,15,19,z,z,z,z,z
9019 DATA p$,18,z,z,z,z,z,z
9020 DATA "all'INFERNO... Un diavolo dice 'Trova l
a direzine giusta e vivrai - altrimenti mor
irai'",30,30,30,21,30,30,2
9021 DATA p$,z,24,22,20,z,z,z
9022 DATA "in una strada senza uscita",z,z,z,21,z,
z,z
9023 DATA "MORTO ALL'INFERNO",z,z,z,z,z,z,
z
9024 DATA "sull'orlo di un baratro profondo",21,25
,z,z,z,z,z
9025 DATA "in una sala bella e piena di gioiell
i",24,z,27,z,z,z,3
9026 DATA "sull'orlo di una profonda voragine. Ci
sono tracce di fuoco e di zolfo qui."
,z,27,z,z,z,23,z
9027 DATA p$,26,z,z,25,z,28,z
9028 DATA p$,z,z,z,z,z,27,18,z
9029 DATA "nuotando in un piccolo lago",z,z,z,
z,z,z,4
9030 DATA "",z,z,z,z,z,z,60
9980 PRINT : PRINT : PRINT : PRINT
9981 PRINT "CONGRATULAZIONI L'avventura e'fini
ta e sei ritornato sano e salvo con tutti i
tesori."
9990 PRINT : PRINT "PUNTEGGIO ";sc: PRINT "Vuoi gi
ocare ancora ?"
9991 LET m$=INKEY$: IF m$="" THEN GO TO 9991
9992 IF m$( TO 1)="y" THEN GO TO 5

```

Suggerimenti per giocare con l'avventura

Suggerimenti per giocare con l'avventura:

(1) Perso nel deserto?

Non disperate, il deserto in realtà non è senza fine e se voi lasciate cadere strategicamente degli oggetti nel deserto, dovrete essere in grado di tracciare una mappa dello stesso. Avrete bisogno di questa mappa per trovare l'oasi.

(2) Non potete passare attraverso la porta del soggiorno?

Avete bisogno della chiave che si trova sepolta da qualche parte nel deserto. Per dissotterrare la chiave avrete, naturalmente, bisogno di una vanga che si trova anch'essa fuori dalla porta, da qualche parte.

(3) Finito all'inferno?

Fate quello che dicono i diavoli.

(4) Non riuscite a oltrepassare il pirata?

Provate il rum.

(5) Non riuscite a uccidere il dragone?

Continuate a tentare

(6) Non trovate la chiave del cancello?

Provate con la corda.

(7) Non potete attraversare il baratro?

Forse una bacchetta magica vi sarebbe d'aiuto.

(8) Non riuscite a trovare l'ultimo tesoro?

Andate a nuoto.

La qualità dei programmi presentati e delle tecniche di programmazione adottate fanno di questo testo un'opera particolare che spicca in mezzo ai non pochi libri di programmi per Spectrum. Un esame attento dei listati consentirà al lettore di apprendere i "segreti" della programmazione strutturata e migliorare notevolmente le proprie capacità di programmatore.

I programmi presentati vanno dagli Arcade più famosi, tra cui il celebre "Spectrum Invaders" ai programmi di utilità più interessanti, dai giochi d'azzardo ai programmi didattici, dai programmi funzionali a quelli di matematica e di giochi di strategia.

178

WILLIAMSON
FENLON
DILLON
SPEER
JUN
P. Williams

GRUPPO
EDITORIALE
JACKSON

